REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)* 17/05/2006	**2. REPORT TYPE** Final Report	**3. DATES COVERED** *(From - To)* 01/01/03 - 12/31/05

4. TITLE AND SUBTITLE Optimal Controller for Turbulent Flow Over an Airfoil	**5a. CONTRACT NUMBER** **5b. GRANT NUMBER** F49620-03-1-0038 **5c. PROGRAM ELEMENT NUMBER**
6. AUTHOR(S) John Kim	**5d. PROJECT NUMBER** **5e. TASK NUMBER** **5f. WORK UNIT NUMBER**

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) The Regents of the University of California 10920 Wilshire Boulevard, Suite 1200 Los Angeles, CA 90024-6502	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) USAF/AFRL AFOSR 875 North Randolph Street Arlington VA 22203	**10. SPONSOR/MONITOR'S ACRONYM(S)** AFRL-SR-AR-TR-06-0417

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release,
distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The objectives of this project are twofold: to develop a detached-eddy simulation (DES) technique for turbulent flows past an airfoil at a high angle of attack, and to explore new control strategies for the separated flow, utilizing modern control theories. A system identification approach is used to construct an approximate linear system model for complex flows. The linear quadratic Gaussian (LQG) control synthesis is then used to design optimal controllers for the identified linear model. A system-identification-LQG approach is applied to control a separated boundary layer flow on a flat plate. The controller design based on the identified linear model is shown to reduce the time-averaged separation bubble size.

15. SUBJECT TERMS

Flow control, separation control, detached-eddy simulation, systems-theory approach, system identification.

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER** *(Include area code)*

FINAL REPORT

OPTIMAL CONTROLLER FOR TURBULENT FLOW OVER AN AIRFOIL

F49620-03-1-0038

John Kim
Department of Mechanical and Aerospace Engineering
University of California, Los Angeles

1 Technical Objectives

The objectives of this project are twofold: to develop a detached-eddy simulation (DES) technique for turbulent flows past an airfoil at a high angle of attack, and to explore new control strategies for the separated flow, utilizing modern control theories.

The ability to control flows to achieve a desired effect is a matter of tremendous consequence in many applications. Not surprisingly, there has been enormous interest in controlling flows to achieve such effects for well over a century. Traditional flow control approaches have been based primarily on the control designers' physical insight into the relevant flow physics together with some simple trial and error. While such approaches have been successful and will continue to play a significant role, the incorporation of model-based control theory into fluid mechanics presents new opportunities for many open problems in fluid mechanics.

The systems theory approach to flow control, in which modern control theories are utilized in designing optimal control inputs, is a new promising approach to flow control in general and turbulence control in particular. While it has been demonstrated that the systems theory approach is indeed a viable and promising approach to controlling simple flows (e.g., turbulent channel flows and boundary layers), extending this approach to complex flows (e.g., flows past an airfoil at an angle of attack) presents many new challenges. For simple flows, a linear model of a flow under consideration can be constructed by using the linearized Navier-Stokes equations, and the resulting linear system is used in control synthesis. Fourier decomposition is typically used to transform the original control problem into a set of decoupled, lower-dimensional systems, thus converting a large linear system into a large number of small systems. Examples of successful applications of systems theory approach to flow control can be found in Lee *et al.* (2001), Högberg (2001), Lim (2003).

Applications of the systems theory approach to complex flows are not straightforward, because the required linear system is not readily available and/or the resulting system is too large to handle. We have developed a new approach to constructing a linear system model for complex flows. Specifically, we used a system identification approach to obtain an approximate linear model. The linear quadratic Gaussian (LQG) control synthesis was then used to design optimal controllers for the identified linear model. A system-identification-LQG approach was applied to control a separated boundary layer flow on a flat plate, where the separation was induced by an imposed adverse pressure gradient on the opposite wall (Huang 2005).

What follows is a summary of technical approaches and accomplishments. Further details are referred to the attached appendix.

2 Technical Approaches

The incompressible flow considered in the present study is governed by the Navier-Stokes equations and the continuity equation, which after proper non-dimensionalization can be written as

$$\frac{\partial v_i}{\partial x_i} = 0 \tag{1}$$

$$\frac{\partial v_i}{\partial t} + \frac{\partial (v_i v_j)}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{1}{\mathsf{Re}} \frac{\partial^2 v_i}{\partial x_j \partial x_j} \tag{2}$$

where v_i is the velocity component in Cartesian coordinate x_i ($i = 1, 2, 3$), p is pressure, and Re is the Reynolds number. It is known that the spectrum of length scales rapidly widens when the Reynolds number is increased. LES generally can reach higher Reynolds numbers higher than DNS, but still requires near-DNS resolution near the wall to obtain accurate results. To compute high-Reynolds-number flows, we used the DES approach, in which the appropriately filtered Navier-Stokes equations and the eddy viscosity transport equation written in generalized coordinates (for complex-geometry flows) are

$$\frac{1}{J} \frac{\partial q_i}{\partial \xi_i} = 0 \tag{3}$$

$$\frac{\partial q_i}{\partial t} + \frac{\gamma^i_m}{J} \frac{\partial}{\partial \xi_j} \left(\frac{c^m_k}{J} q_k q_j \right) = -\alpha^{ij} \frac{\partial p}{\partial \xi_j} + \frac{\gamma^i_k}{J} \left[\left(\frac{1}{\mathsf{Re}} + \nu_T \right) \alpha^{mn} \frac{\partial}{\partial \xi_n} \left(\frac{c^k_\ell q_\ell}{J} \right) \right] + \frac{\alpha^{\ell m} c^k_j}{J} \frac{\partial}{\partial \xi_\ell} \left(\frac{c^j_n q_n}{J} \right) \frac{\partial \nu_T}{\partial \xi_k} \tag{4}$$

$$\frac{\partial \tilde{\nu}}{\partial t} + \frac{1}{J} \frac{\partial}{\partial \xi_i} (q_i \tilde{\nu}) = c_{b1} \tilde{S} \tilde{\nu} - \frac{c_{w1} f_w}{\mathsf{Re}} \left[\frac{\tilde{\nu}}{\tilde{d}} \right]^2 + \frac{1}{\sigma \mathsf{Re}} \left\{ \frac{1}{J} \frac{\partial}{\partial \xi_i} \left[(1 + \tilde{\nu}) \alpha^{ij} \frac{\partial \tilde{\nu}}{\partial \xi_j} \right] + c_{b2} \frac{\alpha^{ij}}{J} \frac{\partial \tilde{\nu}}{\partial \xi_i} \frac{\partial \tilde{\nu}}{\partial \xi_j} \right\} \tag{5}$$

where $c^i_j = \partial x_i / \partial \xi_j$, $\gamma^i_j = J (c^i_j)^{-1}$, $\alpha^{ij} = \gamma^i_\ell \gamma^j_\ell / J$, are the transformation metric terms, J is the Jacobian of transformation, and $q_i = \gamma^i_j u_j$, u_j being the filtered Cartesian velocity components. Equation (5) is is actually the Spalart-Allmaras turbulence model with the following modified definition for $\tilde{d}$:

$$\tilde{d} = \min(d, C_{DES} \Delta) \tag{6}$$

where d is the minimum distance to the wall, Δ is a representative local grid size, and C_{DES} essentially determines the boundary dividing the LES and RANS regions. The eddy viscosity ν_T in (4) and $\tilde{\nu}$ in (5) are related by $\nu_T = f_{v1} \tilde{\nu}$. The definitions of model constants $\{c_{b1}, c_{b2}, c_{w1}, \sigma\}$ and functions $\{f_w, f_{v1}, \tilde{S}\}$ in (5) can be found in Nikitin *et al.* (2000).

A linear system model is required to design optimal control input, which minimizes a certain cost function. In particular, we need a discrete-time state-space model in the following form:

$$x_{i+1} = A x_i + B u_i \;, \tag{7}$$

$$y_i = C x_i + D u_i \;, \tag{8}$$

where (A, B, C, D) are system matrices, x_i the state vector, u_i the control input, and y_i the measurement vector. Here i denotes the time-step index. In LQG-control synthesis, the optimal control sequence u_i can be obtained in the following form

$$u_i = -K \hat{x}_i \;, \tag{9}$$

which minimizes the cost function

$$J = \sum_{i=1}^{\infty} x_i^T R x_i + \gamma u_i^T R u_i \ . \tag{10}$$

In equation (9), $\hat{x}_i$ denotes an estimated state obtained from the Kalman filter,

$$\hat{x}_{i+1} = A\hat{x}_i + Bu_i + L(y_i - C\hat{x}_i - Du_i) \ , \tag{11}$$

which estimates the evolution of the state vector x. For simple flows, the system matrices (A, B, C, D) can be derived easily (*e.g.* Lee *et al.* 2001, Lim 2003), and the control gain K in (9) and the Kalman filter gain L in (11) can be obtained by solving corresponding algebraic Riccati equations, which relate the control (filter) gain matrix with the system matrices.

For complex flows, the system matrices are not readily available. The system matrices are instead estimated from properly-designed input-output data sequences using a system identification technique. We have shown that a subspace system identification method works better than direct least-square estimates of the auto regression with an exogenous input (ARX) model in identifying a flow system with noisy input-output signals (Huang 2005). The subspace identification procedure starts with exciting the flow system with a known actuation sequence u_i. The wall measurement sequences y_i, together with u_i, are used to construct data matrices of the form

$$M = [U_f \ \ U_p \ \ Y_f] \ , \tag{12}$$

where U_f, U_p and Y_f represent future input, past input and future output, respectively. Once M is constructed, it can be used to obtain the associated Hankel matrix, which in turn is used to compute the estimate of system matrices (A, B, C, D). More details of the subspace identification method can be found in Van Overschee & De Moor (1996) and Gibson *et al.* (2000) and the references therein.

3 Accomplishments

We have developed numerical methods and computational codes for solving (3), (4) and (5) implemented on parallel computers and a subspace system identification procedure in Huang (2005). The codes are capable of performing DNS, LES, DES and RANS within a unified framework and have been extensively tested and validated. In particular, our domain decomposition method remove prior problem-size limitations and allows much larger simulations to be performed provided that sufficient number of processors are available. This has put us in a position to focus on turbulence modeling and control design issues. More details are found in Huang (2005).

We applied the the system-identification-LQG-based approach to a separated bubble on a flat plate, which was a simplified model for the leading-edge flow separation on an airfoil. This simplified flow model was used to save computational resources, but the control scheme developed here can be applied to complex airfoil flows without modification. The separation bubble was created by applying an adverse pressure gradient to an incoming Blasius boundary layer. The subspace method gave more accurate results than the direct least-square estimate of an ARX model that we had used earlier (Figure 1). The good agreement among the measurements generated by a state

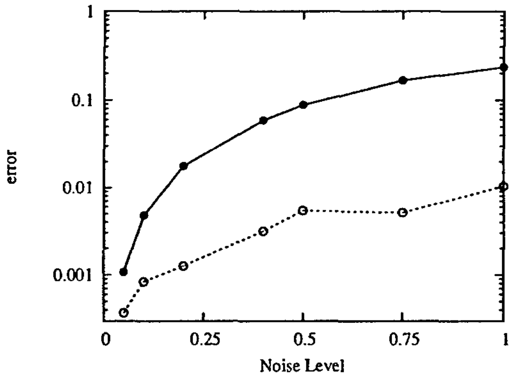

Figure 1: Identification errors. (Solid) Least-square estimate of the ARX model. (Dashed) Subspace identification method.

estimator based on the identified model, its reduced-order model, and the full Navier-Stokes equations (Figure 2) suggested that the approximate models were able to capture important features of the separated flow. The controller, based on the reduced-order model using the LQG synthesis, was then applied to the Navier-Stokes simulations. Figure 3 shows the mean velocity profiles for the controlled and uncontrolled separated boundary layers. The controller design based on the identified linear model was shown to reduce the time-averaged separation bubble size.

Although we have developed a systems theory approach to flow control and have achieved the some favorable control results for the model problem, we have also identified a number of issues which require further investigation, including:

- The optimal approach to excite the system in a multiple-input and multiple-output (MIMO) setting. This is an issue since turbulent flow is a distributed system and some standard techniques developed for lumped systems do not apply.
- An appropriate cost function that can be directly related to reduction of separation bubble size.
- Adjusting control parameters to account for the evolution of the mean flow (this is an issue of gain scheduling).
- The influence of identification and modeling errors to controller performance.

References

[1] Gibson, J. S., Lee, G. H., & Wu, C.-F. 2000. Least-squares estimation of input/output models for distributed linear systems in the presence of noise, *Automatica*, Vol. 36, pp. 1427-1442.

[2] Högberg, H., Bewley, T. R. & Henningson, D. S. 2001. Linear feedback control and estimation of transition in plane channel flow. *J. Fluid Mech.*, Vol. 481 pp. 149-175.

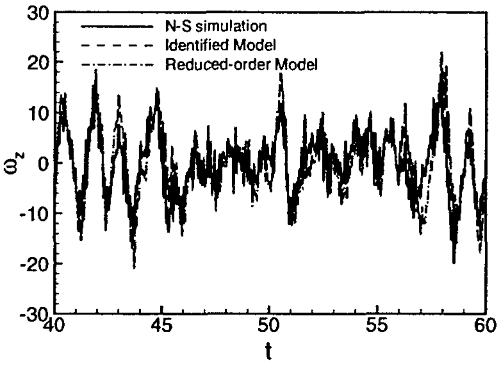

Figure 2: Vorticity fluctuations time history of Navier-Stokes simulations, identified model, and reduced-order model.

[3] Huang, S. C. 2005. Numerical simulation and feedback control of separated flows. Ph.D dissertation, University of California, Los Angeles.

[4] Lee, K., Cortelezzi, L., Kim, J. & Speyer, J. 2001. Application of reduced-order controller to turbulent flows for drag reduction. *Phys. Fluids*, Vol. 13, pp. 1321-1330.

[5] Lim, J. 2003. Control of wall-bounded turbulent shear flows using modern control theory, Ph.D dissertation, University of California, Los Angeles.

[6] Nikitin, N. V., Nicoud, F., Wasistho, B., Squires, K. D. & Spalart, P. R. 2000. An approach to wall modeling in large-eddy simulation. *Phys. Fluids*, Vol. 12, pp. 1629-1632.

[7] Van Overschee, P. & De Moor, B. 1996. Subspace identification for linear systems: theory, implementation, applications, Kluwer Academic Publishers.

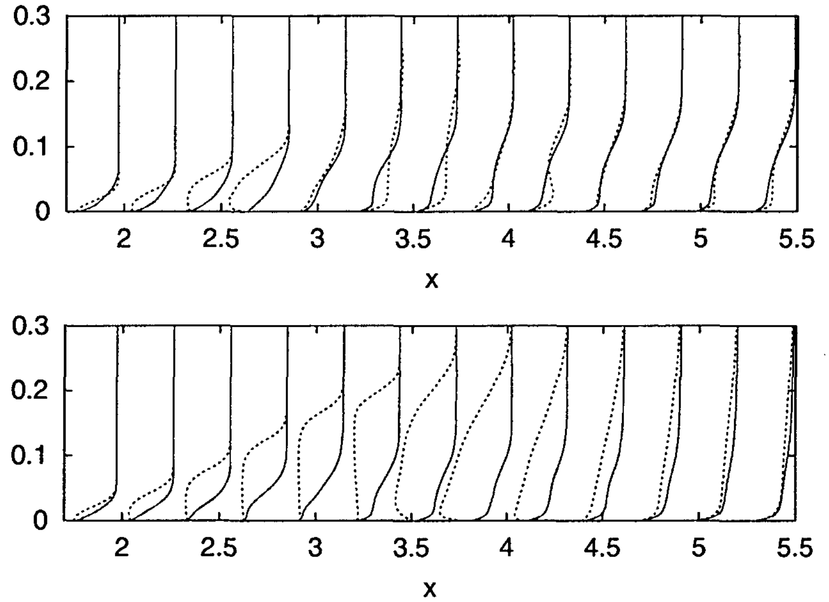

Figure 3: Mean velocity profiles of a separated boundary layer. (top) 2D separated boundary layer. (bottom) 3D separated boundary layer. Solid: controlled. Dashed: uncontrolled.

Personnel Supported During Duration of Grant

Bharanidharan Rajamani,	Graduate Student, University of California, Los Angeles
Shao-Ching Huang,	Graduate Student, University of California, Los Angeles
John Kim,	Professor, University of California, Los Angeles

Publications

"Identification and control of separated boundary layers," Proceedings of the Tenth European Turbulence Conference (eds. Andersson and Krogstad), Trondheim, Norway, June 29-July 2, 2004.

"A singular value analysis of boundary layer control," *Phys. Fluids*, **16**(6), 2004.

"Effects of hydrophobic surface on skin-friction drag," *Phys. Fluids*, **16**(7), 2004.

Honors & Awards Received

Medal for Exceptional Scientific Achievement, NASA (1985)

Fellow, American Physical Society (1989)

Otto Laporte Award, American Physical Society (2001)

Ho-Am Prize in Engineering, Ho-Am Foundation (2002)

AFRL Point of Contact

James Myatt, ARRL/VACA, WPAFB, OH, Phone 937-255-8498.
Carl Tilmann, AFRL/VAAA, WPAFB, OH, Phone 937-255-4077.

Transitions

None.

New Discoveries

None.

APPENDIX

Numerical Simulation and Feedback Control of Separated Flows

Shao-Ching Huang
John Kim

Department of Mechanical and Aerospace Engineering
University of California, Los Angeles
Los Angeles, CA 90095-1597

Abstract

A new closed-loop control approach for separated flows is developed. In order to perform numerical simulations of complex flows with and without control, an efficient flow solver which can treat high-Reynolds number turbulent flows is first developed, and tailored for flow control study. Turbulence simulation techniques including direct numerical simulation, large-eddy simulation, detached-eddy simulation, and Reynolds-averaged Navier-Stokes calculations are incorporated into a unified numerical framework. The algorithm is based on finite-difference discretization of appropriately filtered Navier-Stokes equations and turbulence model equations written in generalized coordinates with a domain decomposition scheme for parallel computation. The new flow solver is validated by comparing computational results against available ones found in the literature. For simple flows, such as turbulent channel and boundary layers, several investigators have successfully designed controllers based on linear optimal control theory. However, applying the same procedure to complex separated flows is hindered owing to the fact that certain required information of the flow is not readily available. In this study, the control of a separated boundary layer exposed to an adverse pressure gradient is considered as a model problem for separated flow past an airfoil at an angle of attack. An approximate linear model of the flow is obtained from input-output data sequences using a subspace system identification method and a model reduction scheme, instead of from linearized Navier-Stokes equations. The linear model is able to capture a number of important features of the separated flow. A linear quadratic Gaussian control synthesis is then used to obtain the optimal feedback control law. Effects of the controller are investigated by comparing controlled flows to uncontrolled ones and those controlled by conventional open-loop schemes. The feedback control is able to reduce the time-averaged separation bubble size.

Chapter 1

Introduction

1.1 Overview

1.1.1 Flow Past an Airfoil

At zero angle of attack, the incoming flow toward the airfoil, referred to as the freestream, splits into two streams at the leading edge and boundary layers form on each side of the airfoil. The disturbances contained in the freestream may grow or decay in the boundary layer, depending on factors including the Reynolds number, surface roughness, surface curvature. The question how incoming small disturbances evolve in the vicinity of the leading edge in the background laminar flow is known as the problem of leading edge receptivity and has been the subject of extensive study [19].

The attached boundary layer near the leading edge is accelerated rapidly and then experiences adverse pressure gradient further downstream. Natural transition may occur due to the viscous Tollmien-Schlichting instability in the boundary layer. At lower Reynolds number, transition can be triggered when artificial tripping is applied or the turbulence intensity in the free stream is high. Transition near the airfoil surface may be delayed or even suppressed by active control (*e.g.,* by suction) such that laminar flow can be sustained for a larger portion of the airfoil than its uncontrolled counterpart. Research along this line is known as the laminar flow control (LFC). Reviews of the LFC technology during 1930's to 1990's can be found in Joslin [45].

When the angle of attack is increased, the flow may separate from the airfoil surface and the flow pattern is quite different from its counterpart of zero angle of attack. There are many possible scenarios for this configuration. When laminar separation occurs on the suction side of the airfoil, the flow may undergo rapid transition to turbulence in the shear layer and the turbulent flow may reattach to the airfoil surface. In this case, a closed (in the time-averaged sense) region is formed and is often referred to as a "laminar separation bubble" [40]. The length of the separation bubble is sensitive to various parameters of the background flow, such as the level of disturbance intensity, pressure gradient and Reynolds number. A "short" bubble can burst into a "long" bubble [71] or break into a completely separated shear layer under different conditions. Much research has been devoted to advance the understanding of laminar separation bubbles in the past few decades. Gaster [28], for example, experimentally studied the mean flow structure of a laminar separation bubble. Pauley *et al.* [72] numerically studied the vortex shedding features of a laminar boundary layer subject to adverse pressure gradient. They applied a suction profile on the top boundary of their computational domain, creating a potential flow field with adverse pressure gradient in the streamwise direction. To understand how disturbances evolve in a laminar separation bubble, Watmuff [93] performed detailed measurement of a laminar separation boundary layer. Impulsive disturbances are introduced into the flowfield at the bottom of the boundary layer. The formation of three-dimensional vortex loops from two-dimensional wave packets was observed and carefully studied. Alam and Sandham [3] carried out direct numerical simulation of laminar separation bubbles. Top-boundary suction is used to create a adverse pressure gradient and time-periodic blowing and suction at the wall is applied to perturb the incoming boundary layer.

At higher Reynolds number, the boundary layer transition may take place before separation. The resulting turbulent boundary layer may separate, remain attached, or separate and then reattach, depending on the Reynolds number, the angle of attack and many other parameters. Separated turbulent boundary layers are quite different from their attached counterparts, and are not well understood [68].

On the pressure side of the airfoil, the (laminar or turbulent) boundary layer typically remains (fully or partially) attached. At the trailing edge, strong interaction among the vortices generated from the suction and pressure sides of the airfoil can occur. The overall flow pattern is rather complicated over a wide range of Reynolds numbers.

When the angle of attack is increased beyond the stall angle, the flow over an airfoil becomes fully separated, resulting in significant loss of the lift force. Flow separation can also be found in many other engineering applications, such as diffusers and turbine cascades operating under off-design conditions. Flow separation can result in the device's loss of efficiency, unfavorable unsteadiness and noise generation. A separated airfoil flow typically consists of freestream, boundary-layer, and wake regions. The flow in the boundary-layer region undergoes a dynamic process, which includes local acceleration/deceleration, separation, and reattachment (if the angle of attack is not too large). Depending upon the conditions at freestream and the geometry of the airfoil, transition to turbulence may take place somewhere during this process. In the wake region, the flow patterns show the features observed in other canonical blunt-body flows, such as shear-layer instability, vortex shedding and mutual interaction of vortices. The wake exhibits self-excited global oscillations without external forcing, but it may also respond to external forcing. These and other related features have important consequences in flow control, and therefore a thorough understanding of the airfoil flow at high angles of attack is an essential step toward developing effective control strategies for this technologically important flow. The complexity of this flow, however, makes both experimental and numerical investigations extremely challenging (see, *e.g.*, [3,16,52,68,86,87,93]).

1.1.2 Active Control of Complex Flow

Active flow control applied to an airfoil at high angles of attack has been a subject of extensive investigations owing to its technological importance. Various control schemes for separated flows have been tried in the past. While the use of passive flow control methods [26,27] require no energy input and has a long history, active control methods have gained much attention in the last decade, partially due to recent progress in the sensor and actuator technology [35].

Active flow control approaches can be loosely divided into two groups: open-loop control and closed-loop control. Most existing open-loop control schemes for a flow over an airfoil have been based on the investigators' physical insight into the flow, and have been a trial-and-error approach. In this approach, the control actuation, such as those of the synthetic-jet type [30] or by sound [18], forcing the flow at certain frequencies (usually related to the vortex shedding or instability mechanisms) is applied in a predetermined fashion without feedback of instantaneous flow state. Although how forcing affects separated flows generally have not been completely understood, some success has been reported in controlling separated flows by choosing appropriate forcing frequency, amplitude and locations (see, *e.g.*, [21,33,79–81]). A number of examples using this approach are briefly reviewed below.

Seifert and Pack [79,81], experimentally studied the effects of oscillatory blowing on the NACA 0015 airfoil at chord Reynolds number of 3.1×10^7. Oscillatory blowing at reduced frequency, normalized by the airfoil chord length and freestream velocity, in the range of 0.5 and 1 was found to be most effective. It was demonstrated that oscillatory actuation significantly alter the lift coefficient for some parameter sets. The momentum coefficient required by oscillatory blowing is orders of magnitude smaller than steady blowing to achieve similar level of aerodynamic performance. Donovan *et al.* [21] numerically investigated the effectiveness of surface jet applied on the NACA 0015 and NACA 0012 airfoils at Reynolds number in the order of $\mathcal{O}(10^6)$. They found that the synthetic jet significantly increases the lift coefficient in the post-stall regime. Hassan *et al.* [34] numerically investigated the effect of an array of synthetic jets on an NACA 0012 airfoil at Reynolds number of 3×10^6 and Mach number of 0.6. The synthetic jet is applied over 10% of NACA 0012 airfoil's surface. They examined the effects of jet locations, peak jet velocity and jet frequency on the lift, drag and pitching moment. Wu *et al.* [95] used the NACA 0012 airfoil to study the effects of various actuation frequencies and amplitudes. Reviews of active separation control can be found in Greenblatt and Wygnanski [33].

In these studies, the two main control parameters are excitation frequency and excitation amplitude. Other key parameters include angle of attack, Reynolds numbers and actuation location. It should be noted that in the cases mentioned above, only a (single) prescribed excitation frequency was applied to the flow. The exploration of some frequency range has to be performed in order to find maximum performance gain (usually lift enhancement or drag reduction). Global flow parameters, such as moment coefficient, lift coefficient, drag coefficient, or their combinations, are the measuring factors to examine the efficiency of the applied control scheme. Instantaneous surface pressure variations are used in some cases to explain the observed performance change with and without control.

It is noted that detailed dynamics in controlled flows is generally unavailable experimentally, probably due to the difficulties of measuring the complex flowfields. As an alternative, computational approach may provide a way to explore the parameter space at (possibly) a lower cost. However, while reasonable lift and/or drag coefficients were obtained in these calculations for small angles of attack, discrepancies among computational results were reported in some cases even without control, especially at post-stall angles of attack [95]. In fact, this situation is not only specific to flow control, but also to separated flows in general. One possibility of these discrepancies may be the use of two-dimensional Reynolds-average Navier-Stokes (RANS) method for computing separated flows.

On the other hand, in a closed-loop control approach, instantaneous flow information (such as pressure or shear stresses) is used to determine the current and future control actuation command. Methods based on modern control theory have recently emerged as a promising approach than conventional trial-and-error methods [9,48,64].

In this control-theoretic approach, the problem of controlling a nonlinear flow is formulated as a nonlinear optimization problem, in which the objective is to minimize a certain cost function, properly defined to achieve certain control goals. General nonlinear optimization techniques have been developed for flow control [1,8], but often come with high computational cost, which makes this approach difficult to apply to real flow systems, especially for three-dimensional turbulent flows. Nevertheless, using such an approach, Bewley *et al.* [11] was able to show that, by choosing an appropriate optimization time horizon and cost functions, it was possible to laminarize a fully-developed turbulent channel flow at $Re_\tau = 100$, where Re_τ is the Reynolds number based on half channel height and the friction velocity at the wall.

A suboptimal approach developed by Lee *et al.* [54] used a very small time horizon (in [54] the time horizon equals the computational time step) and linear adjoint equations to avoid the iterative procedure used in [11]. The approach was applied to turbulent channel flow at $Re_\tau = 100$ and achieved 20% drag reduction. A similar approach was applied to the control of laminar vortex shedding behind a circular cylinder [61], and the control of a turbulent backward-facing step flow [47].

An alternative to the nonlinear optimization approach is to formulate the flow control problem within the framework of linear optimal control theory [32]. In this approach the system dynamics is assumed linear; all nonlinear effects are considered "noise" to the flow system (or the *plant*, in control-theory terminology). One major advantage of using linear optimal control theory is that the state-space solution of the optimal control and efficient and stable computational algorithms have been available [24,96].

Some success using this approach has been reported in the last decade, aiming at suppressing disturbances (*e.g.,* to delay transition) in a laminar channel [6,10,44] or in a laminar boundary layer [37], or reducing skin-friction drag of turbulent channel flows [37,38,55]. Some implications of applying a linear approach to nonlinear wall-bounded flows were discussed by Kim and Lim [49].

The common elements in these control methods are (1) a prescribed mean velocity profile, based on which the control is designed, and (2) spectral decomposition of the system equations. Specifically, the spanwise and streamwise periodicity is used to apply Fourier decomposition so that a multi-dimensional problem is transformed into a series of one-dimensional problems. Therefore, this approach is essentially limited for simple flow configurations.

An interesting development of this approach is that, although the Fourier decomposition based control design is a global approach (*i.e.*, the computation of control command at one point requires information from components of all wavenumbers), it has been shown [5,37] that the control kernel decays exponentially in space. This implies that it may be possible to construct a compact physical-space control kernel for real implementation.

The success of applying the linear optimal control approach for controlling attached boundary layer flows led to the question as to whether it is possible to extend this approach to more complex flows, such as the flow past a pitched airfoil for lift enhancement or separation reduction. For complex flows, however, certain system information required to formulate the control problem using linear optimal control theory is not readily available. A possibility to circumvent this difficulty is to use the system identification techniques [70] to construct an approximate linear model for the flow system using the input-output data sequences [43,76].

1.2 Objectives

Motivated by the success of applying linear control to turbulent flows, the present study aims at developing closed-loop strategies for controlling separated flows. Like some previous studies based on system-theoretic approaches, numerical experiments will be used for performance checks, and to evaluate feasibility. Numerical experiments have the advantages of providing detailed flow dynamics (for both controlled and uncontrolled cases) that allow us to examine the effects of control actuation, as direct numerical simulation (DNS) and large eddy simulation (LES) have been established as reliable tools for turbulence research [65].

However, numerical experiments for complex spatially-developing separated flows (such as flow past an airfoil) are considerably more expensive than for simple flows (*e.g.,* fully developed channel flow). In practice, additional computational cost mainly comes from large mesh size (*i.e.*, the mesh size should be small enough to resolve developing turbulence and the domain size should be large enough to capture large-scale vortical structures), wider space and time spectra (*i.e.*, low-frequency oscillations and high-frequency turbulent fluctuations), and more complicated (hence more expensive) solution algorithm for handling the irregular geometry. The computational cost is even higher if more realistic (higher) Reynolds numbers are considered in simulations. We note that most previous feedback control studies considered flow at very low Reynolds numbers. In addition, the control design procedure is often iterative for controller parameter tuning. Although this is a standard practice and works well for low-dimensional systems, it becomes computationally expensive for the case of flow control (*i.e.*, high-dimensional distributed systems), as each iteration itself takes a significant amount of time to compute. The overall computing cost is high when the number of iterations is increased.

The objectives of this study are: (1) to develop efficient simulation tools for performing reliable simulations of separated flows, and (2) to develop a feedback control strategy for separated flows utilizing linear optimal control theory.

To achieve the first objective, we developed a generalized-coordinate flow solver utilizing massively distributed-memory parallel computers to shorten simulation time. The parallel algorithm essentially removes memory limitation, so that large-scale (*i.e.*, very fine resolution) simulations can be carried out when necessary. To handle high Reynolds number turbulent flows, we use a unified numerical framework to incorporate the DES (Detached-eddy simulation), LES and RANS techniques, so that stand-alone accuracy check can be made to establish the solver's reliability. The developed computational tools are used to study the flow physics of separated flows, and how control affects the flow physics. To achieve the second objective, we use system identification methods to construct approximate linear system model using input-output data and to construct the feedback control using an LQG (Linear Quadratic Gaussian) control synthesis.

Chapter 2

Mathematical Formulations

This chapter summarizes the governing equations for incompressible flow written in generalized coordinates and turbulence models, including those for large-eddy simulation and detached-eddy simulation, used in the present study.

2.1 Governing Equations

The governing equations for incompressible flows are the Navier-Stokes equations and continuity equation. Written in nondimensional form using Cartesian coordinates, they are

$$\frac{\partial u_i}{\partial t} + \frac{\partial u_i u_j}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{1}{Re}\frac{\partial^2 u_i}{\partial x_j \partial x_j}, \tag{2.1}$$

$$\frac{\partial u_j}{\partial x_j} = 0, \tag{2.2}$$

where u_i is the velocity component in the x_i direction, p is pressure, and Re is the Reynolds number.

2.2 Large-eddy Simulation

In large-eddy simulations small eddies are modeled by a subgrid-scale stress (SGS) model while large eddies are computed directly by solving filtered Navier-Stokes equations. Conventionally the filtering operation is expressed as

$$\bar{f}(x) = \int f(x,r)G(r)dr, \tag{2.3}$$

where G is a filter kernel, f is an unfiltered variable, and $\bar{f}$ is a filtered variable. Filtering the momentum equations yields

$$\frac{\partial \bar{u}_i}{\partial t} + \frac{\partial \bar{u}_i \bar{u}_j}{\partial x_j} = -\frac{\partial \bar{p}}{\partial x_i} + \frac{1}{Re}\frac{\partial^2 \bar{u}_i}{\partial x_j \partial x_j} + \frac{\partial \tau_{ij}}{\partial x_j}, \tag{2.4}$$

where τ_{ij} is the subgrid-scale (SGS) stress tensor. Using the eddy-viscosity hypothesis, the SGS stresses can be expressed in terms of the resolved strain rate,

$$\tau_{ij} - \frac{1}{3}\tau_{ii} = 2\nu_T \bar{S}_{ij}, \tag{2.5}$$

where

$$\bar{S}_{ij} = \frac{1}{2}\left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right). \tag{2.6}$$

The filtered momentum equations can be written as

$$\frac{\partial \bar{u}_i}{\partial t} + \frac{\partial(\bar{u}_i\bar{u}_j)}{\partial x_j} = -\frac{\partial \bar{p}}{\partial x_i} + \frac{\partial}{\partial x_j}\left[\left(\frac{1}{Re} + \nu_T\right)\bar{S}_{ij}\right]. \tag{2.7}$$

An SGS eddy-viscosity model is needed to compute ν_T. The Smagorinsky model provides a relation between the eddy viscosity and resolved strain rate and can be written as

$$\nu_T = (C_S\Delta)^2|\bar{S}|, \tag{2.8}$$

where

$$\bar{S} = \sqrt{2\bar{S}_{ij}\bar{S}_{ij}}, \tag{2.9}$$

$$\bar{S}_{ij} = \frac{1}{2}\left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right), \tag{2.10}$$

are computed from the resolved velocity field $\bar{u}_i$. Germano *et al.* [29] proposed a method to evaluate the Smagorinsky constant C_S, which is now known as the dynamic procedure.

2.3 Detached-eddy Simulation

The DES approach was proposed by Spalart *et al.* [84] as a means for simulations of high-Reynolds number turbulent separated flows. At high Reynolds numbers, the ratio of large and small turbulence scales grows like $Re^{9/5}$, and that resolving the near-wall small scales using existing DNS/LES techniques requires extensive computational resources, if not out of reach.

In the formulation of Nikitin *et al.* [69], the Spalart-Allmaras turbulence model (S-A) is used to calculate the eddy viscosity in the near-wall region (referred to as the "RANS region"):

$$\frac{\partial \tilde{\nu}}{\partial t} + \overline{u}_j\frac{\partial \tilde{\nu}}{\partial x_j} = c_{b1}\tilde{S}\tilde{\nu} - c_{w1}f_w\left(\frac{\tilde{\nu}}{d}\right)^2 + \frac{1}{\sigma}\frac{1}{\partial x_k}\left[(\nu + \tilde{\nu})\frac{\partial \tilde{\nu}}{\partial x_k}\right] + \frac{c_{b2}}{\sigma}\frac{\partial \tilde{\nu}}{\partial x_k}\frac{\partial \tilde{\nu}}{\partial x_k}, \tag{2.11}$$

where $\nu_T = \tilde{\nu} f_{v1}$. The model functions are

$$f_{v1} = \chi^3/(\chi^3 + c_{v3}^3), \tag{2.12}$$

$$f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}}, \tag{2.13}$$

$$f_w = g\left[(1 + c_{w3}^6)/(g^6 + c_{w3}^6)\right]^{1/6}, \tag{2.14}$$

$$\tag{2.15}$$

where

$$\chi = \tilde{\nu}/\nu, \tag{2.16}$$

$$g = r + c_{w2}(r^6 - r), \tag{2.17}$$

$$r = \frac{\tilde{\nu}}{\tilde{S}\kappa^2 d^2}, \tag{2.18}$$

$$\tilde{S} = S + \frac{\tilde{\nu}}{\kappa^2 d^2} f_{v2}, \tag{2.19}$$

$$S = \sqrt{2\Omega_{ij}\Omega_{ij}}, \tag{2.20}$$

$$\Omega_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i}\right). \tag{2.21}$$

The model coefficients are:

$$c_{b1} = 0.1355, \tag{2.22}$$

$$c_{b2} = 0.622, \tag{2.23}$$

$$c_{v1} = 7.1, \tag{2.24}$$

$$\sigma = 2/3 \tag{2.25}$$

$$c_{w2} = 0.3, \tag{2.26}$$

$$c_{w3} = 2, \tag{2.27}$$

$$\kappa = 0.41, \tag{2.28}$$

$$c_{w1} = \frac{c_{b1}}{\kappa^2} + \frac{(1 + c_{b2})}{\sigma}. \tag{2.29}$$

In equation (2.11), d is the distance to the nearest wall. In the regions away from the wall (referred to as the "LES region"), the eddy viscosity is calculated in the same way, but with a different definition of d, proportional to the local grid size:

$$d = \max_i \Delta x_i, \tag{2.30}$$

where Δx_i is the local grid size in the i-th coordinate direction. The RANS and LES regions are divided according to the value of $\tilde{d}$

$$\tilde{d} = \min(d, C_{DES}\Delta), \tag{2.31}$$

where $C_{DES} = 0.65$ is used as in [69].

2.4 Coordinate Transformation

In order to handle flow problems with curved solid surface (*e.g.,* flow past an airfoil), the governing equations are transformed into a curvilinear coordinate system so that the boundary coordinate line coincides with the curved surface. Consider smooth mapping function $\mathcal{M}_i$, $i = 1, 2, 3$, between the Cartesian coordinates x_i and the curvilinear coordinates ξ_i $(i = 1, 2, 3)$:

$$x_1 = \mathcal{M}_1(\xi_1, \xi_2, \xi_3), \tag{2.32}$$

$$x_2 = \mathcal{M}_2(\xi_1, \xi_2, \xi_3), \tag{2.33}$$

$$x_3 = \mathcal{M}_3(\xi_1, \xi_2, \xi_3). \tag{2.34}$$

The governing equations (2.1)–(2.2) can be transformed from the x_i-coordinate system to the ξ_i-coordinate system by using the transformation metrics

$$c^i_j = \frac{\partial x_i}{\partial \xi_j} \quad \text{and} \quad b^i_j = \frac{\partial \xi_i}{\partial x_j}, \tag{2.35}$$

for $i = 1, 2, 3$ and $j = 1, 2, 3$. The matrices

$$C = \begin{bmatrix} c^1_1 & c^1_2 & c^1_3 \\ c^2_1 & c^2_2 & c^2_3 \\ c^3_1 & c^3_2 & c^3_3 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} b^1_1 & b^1_2 & b^1_3 \\ b^2_1 & b^2_2 & b^2_3 \\ b^3_1 & b^3_2 & b^3_3 \end{bmatrix} \tag{2.36}$$

are related by $BC = I$, I being a 3-by-3 identity matrix. The Jacobian of transformation is defined to be $J = \det(C)$, where $\det(\cdot)$ is the determinant of a matrix. The Cartesian velocity components u_i and the contravariant velocity components v_i are related by $u_i = c^i_j v_j$. Using the chain rule $\partial/\partial\xi_i = b^j_i \partial/\partial\xi_j$ and defining $q_i = Jv_i$, $\alpha_{ij} = Jb^i_\ell b^j_\ell$, and $d^i_j = c^i_j/J$, the Navier-Stokes equations and the continuity equation, are written in the following forms:

$$\frac{1}{J}\frac{\partial q_i}{\partial \xi_i} = 0, \tag{2.37}$$

$$\begin{aligned} \frac{\partial q_i}{\partial t} + b^i_m \frac{\partial}{\partial \xi_k}\left(d^m_\ell q_k q_\ell\right) = {} & -\alpha_{ik}\frac{\partial p}{\partial \xi_k} \\ & + b^i_k \frac{\partial}{\partial \xi_m}\left[\left(\frac{1}{Re} + \nu_T\right)\alpha_{mn}\frac{\partial}{\partial \xi_n}\left(d^k_\ell q_\ell\right)\right]. \end{aligned} \tag{2.38}$$

Similarly, the S-A eddy viscosity equation is also transformed to generalized coordinates:

$$\begin{aligned} \frac{\partial \tilde{\nu}}{\partial t} + \frac{1}{J}\frac{\partial}{\partial \xi_i}(q_i\tilde{\nu}) = {} & c_{b1}\tilde{S}\tilde{\nu} - \frac{1}{Re}c_{w1}f_w\left(\frac{\tilde{\nu}}{d}\right)^2 \\ & + \frac{1}{\sigma Re}\left\{\frac{1}{J}\frac{\partial}{\partial \xi_i}\left[(1+\tilde{\nu})\alpha_{ij}\frac{\partial \tilde{\nu}}{\partial \xi_j}\right] + c_{b2}\frac{\alpha_{ij}}{J}\frac{\partial \tilde{\nu}}{\partial \xi_i}\frac{\partial \tilde{\nu}}{\partial \xi_j}\right\}. \end{aligned} \tag{2.39}$$

In this study, we consider a simplified version of coordinate transformation:

$$x_1 = \mathcal{M}_1(\xi_1, \xi_2), \tag{2.40}$$

$$x_2 = \mathcal{M}_2(\xi_1, \xi_2), \tag{2.41}$$

$$x_3 = \xi_3. \tag{2.42}$$

The expression of other metric terms are derived accordingly. Although this transformation is less general than the transformation (2.36), it is useful for all geometry considered in this study, ranging from boundary layer flow to flow past an uniform-span airfoil.

2.5 Mesh Generation

An elliptic grid generation method by Hsu and Lee [41] is used to generate curvilinear mesh. In this approach, the coordinates are obtained by numerically solving the nonlinear elliptic equations

$$a\frac{\partial^2 x_i}{\partial \xi_1 \partial \xi_1} + b\frac{\partial^2 x_i}{\partial \xi_1 \partial \xi_2} + c\frac{\partial^2 x_i}{\partial \xi_2 \partial \xi_2} = P_i, \tag{2.43}$$

where a, b, c and P_i are functions of $\partial x_i/\partial \xi_j$ for $i, j = 1, 2$. In this technique, the control function P_i is iteratively adjusted along the boundaries to produce orthogonal mesh near boundaries. Grid orthogonality can be controlled along all boundaries. The details of this grid generation method, including a stable discretization method for equations (2.43), can be found in [41] and are not repeated here. Computational experiments show that this elliptic grid generation method typically requires much longer computation time than the hyperbolic grid generation method (*e.g.*, [17]), but can produce very smooth mesh. As turbulence simulations, especially DNS, typically have stringent requirement for grid quality, the elliptic grid generation technique is used in this study. We noted that this mesh generation method is also used in the DNS study of a turbine passage by Kalitzin *et al.* [46].

Chapter 3

Simulation Methods

In this chapter the numerical methods used to solve the governing equations presented in Chapter 2 are discussed. The main purpose is to establish computational tools for studying separated flow physics and flow control. The efficiency of computational codes is of critical importance in order to make controller synthesis and tuning feasible.

This chapter is divided into four sections, which discuss the spatial discretization schemes, time advancement algorithm and the implementation on distributed-memory parallel computers.

3.1 Spatial Discretization

Spatial discretization plays a critical role in turbulence simulations using finite difference methods. Computations using central difference schemes, which lack a numerical dissipation mechanism like upwind schemes, tend to "blow up" in an under-resolved simulation. On the other hand, upwind schemes are stable but may overly suppress the small-scale turbulence fluctuations [14, 63]. It has been a subject of debate whether upwind difference schemes should be used in turbulence simulations, LES in particular [63, 75, 89]. The debate mainly evolves around under-resolved LES. For well-resolved simulations, good results of turbulence statistics have been reported using either central or upwind schemes.

When central difference schemes are used, a number of different approaches to stabilize the computations have been proposed and used, including applying low-pass filters [20, 92], adding numerical dissipation [77], or formulating the scheme such that it discretely conserves the kinetic energy [66]. Continuously applying low-pass filtering extracts energy from the flow, and filtering operation itself introduces arbitrariness to the computations. It should be mentioned that the dealiasing operation [12, 15] in spectral methods also removes energy from the flow. Likewise, numerical dissipation is typically introduced in an *ad hoc* manner [88]. Recently, the energy-conserving formulation has gained some attention because it neither requires additional filtering nor requires adding artificial dissipation to stabilize the simulations [66, 73, 90]. However, in reality, it is difficult to enforce the energy-conserving conditions exactly on non-orthogonal, stretched grids commonly found in complex geometry without sacrificing accuracy, or when the discrete Poisson equation is solved only approximately (as opposed to solving it exactly to within machine accuracy). In addition, how energy redistributes, in the discrete sense, among the resolved wavenumbers is unknown. Nevertheless, good turbulence statistics have been reported using these schemes [59, 66, 73]. In the present study, a second-order central scheme, a third-order upwind scheme and a fourth-order central scheme are used to approximate the spatial derivatives in the Navier-Stokes equations. On Cartesian uniform mesh, the present second-order scheme is kinetic-energy conserving (when the Poisson equation is solved exactly), while the fourth-order scheme is not (although it is non-dissipative). The third-order upwind scheme is always dissipative.

Despite the good properties of energy-conserving schemes for the momentum equations, in our preliminary simulations unacceptable numerical oscillations were found in the eddy viscosity fields of the Spalart-Allmaras turbulence model on non-orthogonal, skewed grids when central differencing is used. Negative eddy viscosity from the oscillating solution eventually caused numerical instability and produced meaningless results. Therefore, as a compromise, upwind schemes are used to discretize the eddy viscosity transport equation of the Spalart-Allmaras turbulence model. Further details of spatial discretization are given below, using the standard staggered grid arrangement [25].

The second-order central-difference scheme is used to approximate all spatial derivatives in the momentum equations. The second-order central-difference scheme is defined as

$$\left.\frac{\partial f}{\partial \xi}\right|_{\xi=\xi_i} \approx \left.\frac{\delta f}{\delta \xi}\right|_i = \frac{f_{i+1/2} - f_{i-1/2}}{\Delta \xi}, \tag{3.1}$$

to approximate the first derivative of f with respect to ξ at grid point indexed by i. Second derivatives are approximated by

$$\left.\frac{\partial}{\partial \xi}\left(\alpha \frac{\partial f}{\partial \xi}\right)\right|_{\xi=\xi_i} \approx \left.\frac{\delta}{\delta \xi}\left(\alpha \frac{\delta f}{\delta \xi}\right)\right|_i = \frac{\left(\alpha \frac{\delta f}{\delta \xi}\right)_{i+\frac{1}{2}} - \left(\alpha \frac{\delta f}{\delta \xi}\right)_{i-\frac{1}{2}}}{\Delta \xi}, \tag{3.2}$$

where

$$\left(\alpha \frac{\delta f}{\delta \xi}\right)_i = \frac{\alpha_i \left(f_{i+\frac{1}{2}} - f_{i-\frac{1}{2}}\right)}{\Delta \xi}. \tag{3.3}$$

The transformation metric terms are approximated in the similar way:

$$\left.\frac{\delta x}{\delta \xi}\right|_i = \frac{x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}}{\Delta \xi}. \tag{3.4}$$

Explicit third-order and fourth-order finite difference methods are also implemented to discretize the convection terms. The fourth-order central difference method to approximate the first derivative is

$$\frac{\delta \phi}{\delta \xi} = \frac{\phi_{i+2} + 8\phi_{i+1} - 8\phi_{i-1} + \phi_{i-2}}{12\Delta \xi}. \tag{3.5}$$

The third-order upwind scheme is

$$\frac{\delta f}{\delta \xi} = \begin{cases} \dfrac{4f_{i+1} + 6f_i - 12f_{i-1} + 2f_{i-2}}{12\Delta \xi}, & \text{if } q_i > 0, \\ \dfrac{-2f_{i+2} + 12f_{i+1} - 6f_i - 4f_{i-1}}{12\Delta \xi}, & \text{if } q_i < 0. \end{cases} \tag{3.6}$$

One-sided finite-difference methods are used near the boundaries. When the second-order scheme is used to approximate the convection terms, the metric terms are computed using the same discretization scheme. When the third-order or the fourth-order scheme is used to approximate the convection terms, the metric terms are computed using a fourth-order scheme. More details of computing the metric terms are given in Appendix A.

When the fourth-order scheme is used for the convection terms, the viscous terms are discretized with a second order scheme. So the formal order of accuracy of the scheme is still second order. Nevertheless, significant accuracy improvement is found when the fourth-order scheme is used. However, in turbulent flow calculations, the fourth-order scheme is found to be unstable when the flowfield is not well-resolved. Under the same conditions, the second-order and the upwind schemes are found to be stable.

3.2 Temporal Discretization

The time advancement scheme uses the Crank-Nicolson method [88] for terms computed implicitly and a low-storage third-order Runge-Kutta method [85] for terms computed explicitly. The fractional step method [2, 23] is used to solve for velocity and pressure in a sequential manner. The discrete continuity and momentum equations are written as

$$\frac{q_i^\ell - q_i^{\ell-1}}{\Delta t} = -2\beta_\ell F_i^{\ell-1} + \beta_\ell(M_i^\ell + M_i^{\ell-1}) + \gamma_\ell E_i^{\ell-1} + \zeta_\ell E_i^{\ell-2}, \tag{3.7}$$

$$D(q^\ell) = 0, \tag{3.8}$$

where E_i and M_i denote terms computed explicitly and implicitly, respectively, F_i the pressure gradient terms and D the discrete divergence operator defined as

$$D(q) = \frac{1}{J}\left(\frac{\delta q_1}{\delta \xi_1} + \frac{\delta q_2}{\delta \xi_2}\right) + \frac{\delta q_3}{\delta \xi_3}. \tag{3.9}$$

The Runge-Kutta substep index $\ell = 0$ corresponds to the flow field at time t and $\ell = 3$ responds to the flow field at time $t + \Delta t$. The coefficients of the Runge-Kutta scheme are:

$$(\beta_1, \beta_2, \beta_3) = \left(\frac{4}{15}, \frac{1}{15}, \frac{1}{6}\right),$$

$$(\gamma_1, \gamma_2, \gamma_3) = \left(\frac{8}{15}, \frac{5}{12}, \frac{3}{4}\right),$$

$$(\zeta_1, \zeta_2, \zeta_3) = \left(0, -\frac{17}{60}, -\frac{5}{12}\right).$$

The terms computed implicitly are

$$\begin{aligned} M_1 &= b_1^1 \frac{\delta}{\delta \xi_2}\left[-d_1^1 q_1 q_2 + \alpha_{22}\nu_t \frac{\delta(d_1^1 q_1)}{\delta \xi_2}\right] \\ &\quad + b_2^1 \frac{\delta}{\delta \xi_2}\left[-d_1^2 q_1 q_2 + \alpha_{22}\nu_t \frac{\delta(d_1^2 q_1)}{\delta \xi_2}\right], \end{aligned} \tag{3.10}$$

$$\begin{aligned} M_2 &= b_1^2 \frac{\delta}{\delta \xi_2}\left[-d_2^1 q_2 q_2 + \alpha_{22}\nu_t \frac{\delta(d_2^1 q_2)}{\delta \xi_2}\right] \\ &\quad + b_2^2 \frac{\delta}{\delta \xi_2}\left[-d_2^2 q_2 q_2 + \alpha_{22}\nu_t \frac{\delta(d_2^2 q_2)}{\delta \xi_2}\right], \end{aligned} \tag{3.11}$$

$$M_3 = \frac{1}{J}\frac{\delta}{\delta \xi_2}\left[-q_2 q_3 + \alpha_{22}\nu_t \frac{\delta q_3}{\delta \xi_2}\right], \tag{3.12}$$

where

$$\nu_t = \frac{1}{Re} + \nu_T, \tag{3.13}$$

has been used to simplify notation. In equation (3.11), M_2 contains terms nonlinear in q_2, and requires an iterative procedure when inverting the implicit system of equations. Alternatively, without loss of overall temporal accuracy, it can be linearized as

$$(q_2 q_2)^\ell = 2q_2^{\ell-1} q_2^\ell - q_2^{\ell-1} q_2^{\ell-1} + \mathcal{O}(\Delta t^2), \tag{3.14}$$

requiring no iterations. The rest of the terms in the Navier-Stokes equations are computed explicitly, expressed as

$$E_i = -E_{Hi} + E_{Vi} + E_{Si} \quad \text{for } i = 1, 2, 3, \tag{3.15}$$

where

$$\begin{aligned} E_{H1} = {} & b_1^1 \frac{\delta}{\delta\xi_1} \left(d_1^1 q_1 q_1 + d_2^1 q_1 q_2\right) + b_1^1 \frac{\delta}{\delta\xi_2} \left(d_2^1 q_2 q_2\right) \\ & + b_2^1 \frac{\delta}{\delta\xi_1} \left(d_1^2 q_1 q_1 + d_2^2 q_1 q_2\right) + b_2^1 \frac{\delta}{\delta\xi_2} \left(d_2^2 q_2 q_2\right) + \frac{\delta}{\delta\xi_3} \left(q_1 q_3\right), \end{aligned} \tag{3.16}$$

$$\begin{aligned} E_{H2} = {} & b_1^2 \frac{\delta}{\delta\xi_1} \left(d_1^1 q_1 q_1 + d_2^1 q_1 q_2\right) + b_1^2 \frac{\delta}{\delta\xi_2} \left(d_1^1 q_1 q_2\right) \\ & + b_2^2 \frac{\delta}{\delta\xi_1} \left(d_1^2 q_1 q_1 + d_2^2 q_1 q_2\right) + b_2^2 \frac{\delta}{\delta\xi_2} \left(d_1^2 q_1 q_2\right) + \frac{\delta}{\delta\xi_3} \left(q_1 q_3\right), \end{aligned} \tag{3.17}$$

$$E_{H3} = \frac{1}{J} \frac{\delta}{\delta\xi_1} \left(q_1 q_3\right) + \frac{\delta}{\delta\xi_3} \left(q_3 q_3\right), \tag{3.18}$$

$$\begin{aligned} E_{V1} = {} & b_1^1 \frac{\delta}{\delta\xi_1} \left[\alpha_{11} \nu_t \frac{\delta}{\delta\xi_1} (d_1^1 q_1 + d_2^1 q_2) + \alpha_{12} \nu_t \frac{\delta}{\delta\xi_2} (d_1^1 q_1 + d_2^1 q_2)\right] \\ & + b_1^1 \frac{\delta}{\delta\xi_2} \left[\alpha_{12} \nu_t \frac{\delta}{\delta\xi_1} (d_1^1 q_1 + d_2^1 q_2) + \alpha_{22} \nu_t \frac{\delta}{\delta\xi_2} (d_2^1 q_2)\right] \\ & + b_2^1 \frac{\delta}{\delta\xi_1} \left[\alpha_{11} \nu_t \frac{\delta}{\delta\xi_1} (d_1^2 q_1 + d_2^2 q_2) + \alpha_{12} \nu_t \frac{\delta}{\delta\xi_2} (d_1^2 q_1 + d_2^2 q_2)\right] \\ & + b_2^1 \frac{\delta}{\delta\xi_1} \left[\alpha_{12} \nu_t \frac{\delta}{\delta\xi_1} (d_1^2 q_1 + d_2^2 q_2) + \alpha_{22} \nu_t \frac{\delta}{\delta\xi_2} (d_2^2 q_2)\right] \\ & + \frac{\delta}{\delta\xi_3} \left(\nu_t \frac{\delta q_1}{\delta\xi_3}\right), \end{aligned} \tag{3.19}$$

$$\begin{aligned} E_{V2} = {} & b_1^2 \frac{\delta}{\delta\xi_1} \left[\alpha_{11} \nu_t \frac{\delta}{\delta\xi_1} (b_1^1 q_1 + b_2^1 q_2) + \alpha_{12} \nu_t \frac{\delta}{\delta\xi_2} (b_1^1 q_1 + b_2^1 q_2)\right] \\ & + b_1^2 \frac{\delta}{\delta\xi_1} \left[\alpha_{12} \nu_t \frac{\delta}{\delta\xi_1} (b_1^1 q_1 + b_2^1 q_2) + \alpha_{22} \nu_t \frac{\delta}{\delta\xi_2} (b_1^1 q_1)\right] \\ & + b_2^2 \frac{\delta}{\delta\xi_1} \left[\alpha_{11} \nu_t \frac{\delta}{\delta\xi_1} (b_1^2 q_1 + b_2^2 q_2) + \alpha_{12} \nu_t \frac{\delta}{\delta\xi_2} (b_1^2 q_1 + b_2^2 q_2)\right] \\ & + b_2^2 \frac{\delta}{\delta\xi_2} \left[\alpha_{12} \nu_t \frac{\delta}{\delta\xi_1} (b_1^2 q_1 + b_2^2 q_2) + \alpha_{22} \nu_t \frac{\delta}{\delta\xi_2} (b_1^2 q_1 + b_2^2 q_2)\right] \\ & + \frac{\delta}{\xi_3} \left[\nu_t \frac{\delta q_2}{\delta\xi_3}\right], \end{aligned} \tag{3.20}$$

$$\begin{aligned} E_{V3} = &\frac{1}{J}\frac{\delta}{\delta\xi_1}\left(\alpha_{11}\nu_t\frac{\delta q_3}{\delta\xi_1}\right) + \frac{1}{J}\frac{\delta}{\delta\xi_1}\left(\alpha_{12}\nu_t\frac{\delta q_3}{\delta\xi_2}\right) \\ &+ \frac{1}{J}\frac{\delta}{\delta\xi_2}\left(\alpha_{12}\nu_t\frac{\delta q_3}{\delta\xi_1}\right) + \frac{\delta}{\delta\xi_3}\left(\nu_t\frac{\delta q_3}{\delta\xi_3}\right), \end{aligned} \tag{3.21}$$

$$\begin{aligned} E_{S1} = &\alpha_{11}\left(b_1^1\frac{\delta\nu_t}{\delta\xi_1} + b_1^2\frac{\delta\nu_t}{\delta\xi_2}\right)\frac{\delta}{\delta\xi_1}\left(d_1^1 q_1 + d_2^1 q_2\right) \\ &+ \alpha_{11}\left(b_2^1\frac{\delta\nu_t}{\delta\xi_1} + b_2^2\frac{\delta\nu_t}{\delta\xi_2}\right)\frac{\delta}{\delta\xi_1}\left(d_1^2 q_1 + d_2^2 q_2\right) \\ &+ \alpha_{12}\left(b_1^1\frac{\delta\nu_t}{\delta\xi_2} + b_1^2\frac{\delta\nu_t}{\delta\xi_2}\right)\frac{\delta}{\delta\xi_1}\left(d_1^1 q_1 + d_2^1 q_2\right) \\ &+ \alpha_{12}\left(b_2^1\frac{\delta\nu_t}{\delta\xi_2} + b_2^2\frac{\delta\nu_t}{\delta\xi_2}\right)\frac{\delta}{\delta\xi_1}\left(d_1^2 q_1 + d_2^2 q_2\right) \\ &+ \alpha_{11}\frac{\delta q_3}{\delta\xi_1}\frac{\delta\nu_t}{\delta\xi_3} + \alpha_{12}\frac{\delta q_3}{\delta\xi_2}\frac{\delta\nu_t}{\delta\xi_3}, \end{aligned} \tag{3.22}$$

$$\begin{aligned} E_{S2} = &\alpha_{12}\left(b_1^1\frac{\delta\nu_t}{\delta\xi_1} + b_1^2\frac{\delta\nu_t}{\delta\xi_2}\right)\frac{\delta}{\delta\xi_1}\left(d_1^1 q_1 + d_2^1 q_2\right) \\ &+ \alpha_{12}\left(b_2^1\frac{\delta\nu_t}{\delta\xi_1} + b_2^2\frac{\delta\nu_t}{\delta\xi_2}\right)\frac{\delta}{\delta\xi_1}\left(d_1^2 q_1 + d_2^2 q_2\right) \\ &+ \alpha_{22}\left(b_1^1\frac{\delta\nu_t}{\delta\xi_2} + b_1^2\frac{\delta\nu_t}{\delta\xi_2}\right)\frac{\delta}{\delta\xi_1}\left(d_1^1 q_1 + d_2^1 q_2\right) \\ &+ \alpha_{22}\left(b_2^1\frac{\delta\nu_t}{\delta\xi_2} + b_2^2\frac{\delta\nu_t}{\delta\xi_2}\right)\frac{\delta}{\delta\xi_1}\left(d_1^2 q_1 + d_2^2 q_2\right) \\ &+ \alpha_{12}\frac{\delta q_3}{\delta\xi_1}\frac{\delta\nu_t}{\delta\xi_3} + \alpha_{22}\frac{\delta q_3}{\delta\xi_2}\frac{\delta\nu_t}{\delta\xi_3}, \end{aligned} \tag{3.23}$$

$$\begin{aligned} E_{S3} = &(b_1^1\frac{\delta\nu_t}{\delta\xi_1} + b_1^2\frac{\delta\nu_t}{\delta\xi_2})\left(d_1^1\frac{\delta q_1}{\delta\xi_3} + d_2^1\frac{\delta q_2}{\delta\xi_3}\right) \\ &+ (b_1^1\frac{\delta\nu_t}{\delta\xi_1} + d_2^2\frac{\delta\nu_t}{\delta\xi_2})\left(d_1^1\frac{\delta q_1}{\delta\xi_3} + d_2^1\frac{\delta q_2}{\delta\xi_3}\right), \end{aligned} \tag{3.24}$$

The pressure gradient terms are $F_i = G_i(p)$, where

$$G_1(\cdot) = \alpha_{11}\frac{\delta\cdot}{\delta\xi_1} + \alpha_{12}\frac{\delta\cdot}{\delta\xi_2}, \tag{3.25}$$

$$G_2(\cdot) = \alpha_{12}\frac{\delta\cdot}{\delta\xi_1} + \alpha_{22}\frac{\delta\cdot}{\delta\xi_2}, \tag{3.26}$$

$$G_3(\cdot) = \frac{\delta\cdot}{\delta\xi_3}. \tag{3.27}$$

The systems of equations to be solved at each Runge-Kutta substeps are summarized below. First, the right hand sides that contain all terms computed explicitly are computed first:

$$R_1 = q_1^{\ell-1} - 2\Delta t\beta_\ell F_1^{\ell-1} + \beta_\ell\Delta t M_1^{\ell-1} + \gamma_\ell\Delta t E_1^{\ell-1} + \zeta_\ell\Delta t E_1^{\ell-2}, \tag{3.28}$$

$$R_2 = q_2^{\ell-1} - 2\Delta t\beta_\ell F_2^{\ell-1} + \beta_\ell\Delta t \tilde{M}_2^{\ell-1} + \gamma_\ell\Delta t E_2^{\ell-1} + \zeta_\ell\Delta t E_2^{\ell-2}, \tag{3.29}$$

$$R_3 = q_3^{\ell-1} - 2\Delta t\beta_\ell F_3^{\ell-1} + \beta_\ell\Delta t M_3^{\ell-1} + \gamma_\ell\Delta t E_3^{\ell-1} + \zeta_\ell\Delta t E_3^{\ell-2}, \tag{3.30}$$

where $\tilde{M_2}$ is obtained using equations (3.11) and (3.14). Then the implicit systems of equations are formed and inverted directly to obtain intermediate velocity fields:

$$\Big\{I + \beta_\ell \Delta t \Big[b_1^1 \frac{\delta}{\delta\xi_2}\left(d_1^1 \hat{q}_2 - \alpha_{22}\nu_t \frac{\delta d_1^1 \cdot}{\delta\xi_2}\right) + b_2^1 \frac{\delta}{\delta\xi_2}\left(d_1^2 \hat{q}_2 - \alpha_{22}\nu_t \frac{\delta d_1^2 \cdot}{\delta\xi_2}\right)\Big]\Big\} \hat{q}_1 = R_1, \quad (3.31)$$

$$\Big\{I + \beta_\ell \Delta t \Big[b_1^1 \frac{\delta}{\delta\xi_2}\left(2d_1^1 {\hat{q}_2}^{\ell-1} - \alpha_{22}\nu_t \frac{\delta d_1^1 \cdot}{\delta\xi_2}\right) + b_2^1 \frac{\delta}{\delta\xi_2}\left(2d_1^2 {\hat{q}_2}^{\ell-1} - \alpha_{22}\nu_t \frac{\delta d_1^2 \cdot}{\delta\xi_2}\right)\Big]\Big\} \hat{q}_2 = R_2, \quad (3.32)$$

$$\Big\{I + \beta_\ell \Delta t \Big[b_1^1 \frac{\delta}{\delta\xi_2}\left(d_1^1 \hat{q}_2 - \alpha_{22}\nu_t \frac{\delta d_1^1 \cdot}{\delta\xi_2}\right) + b_2^1 \frac{\delta}{\delta\xi_2}\left(d_1^2 \hat{q}_2 - \alpha_{22}\nu_t \frac{\delta d_1^2 \cdot}{\delta\xi_2}\right)\Big]\Big\} \hat{q}_3 = R_3. \quad (3.33)$$

Equation (3.32) has to be solved prior to equations (3.31) and (3.33) so that $\hat{q}_2$ is available for the left-hand-side operators. Next, the Poisson equation,

$$\frac{1}{J}\frac{\delta}{\delta\xi_m}\left(\alpha_{mn}\frac{\delta\phi}{\partial\xi_n}\right) + \frac{\delta^2\phi}{\delta\xi_3^2} = \frac{1}{2\beta_\ell\Delta t}\left(\frac{1}{J}\frac{\delta\hat{q}_m}{\delta\xi_m} + \frac{\delta\hat{q}_3}{\delta\xi_3}\right), \quad (3.34)$$

where $1 \le m \le 2$ and $1 \le n \le 2$ is solved to obtain the pressure update $\phi = p^\ell - p^{\ell-1}$, which is then used to obtain the divergence-free field $\mathbf{q}^\ell$:

$$q_i^\ell = \hat{q}_i^\ell - 2\Delta t \beta_\ell G_i(\phi). \quad (3.35)$$

The pressure $p^\ell = p^{\ell-1} + \phi$ is computed once ϕ is available. The details of solving equation (3.34) is given in Section 3.3. The implications of the implicit treatment along the ξ_2 direction on the time step size and on the parallelization on distributed-memory parallel computers are further discussed in Section 3.5.

3.3 Solution Methods of Poisson equation

At each Runge-Kutta substep the discrete Poisson equation to be solved can be written in the form

$$\frac{1}{J}\frac{\delta}{\delta\xi_m}\left(\alpha_{mn}\frac{\delta\phi}{\partial\xi_n}\right) + \frac{\delta^2\phi}{\delta\xi_3^2} = b, \quad (3.36)$$

where

$$b = \frac{1}{\beta_i\Delta t}\left(\frac{1}{J}\frac{\delta\hat{q}_m}{\delta\xi_m} + \frac{\delta\hat{q}_3}{\delta\xi_3}\right), \quad (3.37)$$

and $1 \le m \le 2$ and $1 \le n \le 2$. The Runge-Kutta time stepping index ℓ is suppressed for notational simplicity. This equation is modified at the boundaries where velocity boundary conditions are prescribed, so as to be consistent with the formulation of the fractional step method [50].

The computation of the solution to the Poisson equation at each time step contributes significantly to the overall execution time, so it is important to ensure that the solution procedure is efficient to avoid excessive computation time. Two methods for solving (3.36) are used in this study, depending on the mesh type in the other two directions. When a curvilinear mesh is used (*e.g.,* flow past an airfoil), a cell-centered multigrid method is used to solve equation (3.36). When a Cartesian mesh is used and one of the directions has uniform grid distribution (*e.g.,* flat-plate boundary layer flows), a fast transform method is used. When solving the Poisson equation iteratively, it is not possible to drive the residual of iteration to machine zero. Therefore, the continuity equation is not satisfied exactly. Fortunately, the multigrid solver is able drive the residual to a low enough level efficiently that the computed flowfields become essentially independent of this residual level. When the fast transform method is used, the Poisson equation is directly solved so that the divergence-free condition is satisfied to machine accuracy. More details of these two methods are given separately in the following sections.

3.3.1 Multigrid Method

In this method ϕ and b are first expressed in terms of their discrete Fourier transform components in the ξ_3 direction

$$\phi = \sum_{k=-N_3/2}^{N_3/2-1} \widehat{\phi}_k e^{-2\pi i k \xi_3 / L_3}, \tag{3.38}$$

$$b = \sum_{k=-N_3/2}^{N_3/2-1} \widehat{b}_k e^{-2\pi i k \xi_3 / L_3}, \tag{3.39}$$

where L_3 and N_3 are the domain size and the number of grid points, respectively, in the spanwise direction. Substituting equations (3.38) and (3.39) into (3.36) yields

$$\frac{1}{J}\frac{\delta}{\delta \xi_m}\left(\alpha_{mn}\frac{\delta \widehat{\phi}_k}{\delta \xi_n}\right) - k'\widehat{\phi}_k = \widehat{b}_k, \tag{3.40}$$

for each wavenumber k, where the modified wave number [25] is defined to be

$$k' = \frac{2\left[\cos\left(\frac{2\pi k \xi_3}{L_z} - 1\right)\right]}{\Delta \xi_3^2}. \tag{3.41}$$

Equation (3.40) corresponding to each modified wavenumber is a Helmholtz equation, which can be discretized and written in the standard matrix form

$$A\eta = r, \tag{3.42}$$

where the coefficient matrix A is has nine diagonals. The length of the solution vector η is $N_1 N_2$, where N_1 and N_2 are the numbers of grid cells along the ξ_1 and ξ_2 directions, respectively.

For each wavenumber k, a system of equations of the form of equation (3.42) is solved iteratively using a cell-centered multigrid method with a line iterative method applied along the ξ_2-direction as the smoother. The convergence criterion of the multigrid iteration is defined to be

$$\frac{\|A\eta - r\|}{\|r\|} < \varepsilon, \tag{3.43}$$

where $\|\cdot\|$ is the vector 2-norm [31] and ε is a prescribed tolerance. In the multigrid cycle N_L grid levels are defined with $L = 1$ and $L = N_L$ corresponding to the finest and coarsest grid levels, respectively. Other standard details of the multigrid method can be found in the literature (*e.g.*, see [13]) and are not repeated here. Either the V-cycle or the W-cycle multigrid iterations is used in the computations, depending on which had a shorter total execution time. For most computational cases considered here, the iterations using the W-cycle appeared to converge faster than the V-cycle in terms of total execution time. Once the solution of $\widehat{\phi}_k$ for all wavenumbers are obtained, the solution ϕ in the (ξ_1, ξ_2, ξ_3)-space is calculated by taking the inverse discrete Fourier transform

$$\phi = \sum_{k=1}^{N_3} \widehat{\phi}_k e^{2\pi i k \xi_3 / L_z}. \tag{3.44}$$

3.3.2 Fast Transform Method

When the mesh is uniform along two directions, the fast transform method [50] can be used. Under this condition, the discrete Poisson equation is simplified to

$$\frac{\alpha_{11}}{J}\frac{\delta}{\delta\xi_1}\left(\frac{\delta\widehat{\phi}_k}{\delta\xi_1}\right) + \frac{1}{J}\frac{\delta}{\delta\xi_2}\left(\alpha_{22}\frac{\delta\widehat{\phi}_k}{\delta\xi_2}\right) + \frac{\delta^2\phi}{\delta\xi_3^2} = b. \tag{3.45}$$

Expanding ϕ and b in terms of their Fourier components in ξ_1 and ξ_3 directions yields

$$\phi = \sum_{k_x=-N_1/2+1}^{N_1/2} \sum_{k_z=1}^{N_3} \widehat{\phi}_{k_x,k_z} e^{-2\pi i (k_x\xi_1/L_1 + k_z\xi_3/L_3)}, \tag{3.46}$$

$$b = \sum_{k_x=-N_1/2+1}^{N_1/2} \sum_{k_z=1}^{N_3} \widehat{b}_{k_x,k_z} e^{-2\pi i (k_x\xi_1/L_1 + k_z\xi_3/L_3)}, \tag{3.47}$$

where L_1 is the streamwise domain size, L_3 the spanwise domain size, N_1 the number of grid points along the streamwise direction, and N_3 the number of grid points along the spanwise direction. Substituting (3.46) and (3.47) into (3.45) gives $N_1 \times N_3$ decoupled tridiagonal matrix equations, each of which has the form

$$a_{1,i-1}\widehat{\phi}_{i-1} + a_{2,i}\widehat{\phi}_i + a_{3,i+1}\widehat{\phi}_{i+1} = b_i, \quad i = 1, \cdots, N_2 - 1 \tag{3.48}$$

where i is the grid point index along the ξ_2 direction for each wavenumber pair (k_x, k_z). The tridiagonal systems are inverted directly, followed by a double inverse discrete Fourier transform to obtain

$$\phi = \sum_{k_x=-N_1/2+1}^{N_1/2} \sum_{k_z=1}^{N_3} \widehat{\phi}_{k_x,k_z} e^{2\pi i (k_x\xi_1/L_1 + k_z\xi_3/L_3)}. \tag{3.49}$$

Since the coefficients $a_{1,i-1}$, $a_{2,i}$, and $a_{3,i+1}$ in equation (3.48) are time-invariant, they are computed and LU-factorized at the beginning of the simulation and used throughout the simulations.

3.4 Eddy-viscosity Transport Equation

To solve the eddy-viscosity equation (2.39) numerically, the spatial differentiation operators are first approximated using the upwind schemes described in section 3.1. In order to prevent the temporal stability of the discrete eddy-viscosity equation from affecting the temporal stability of the discrete momentum equations, implicit Crank-Nicolson method is used for the temporal discretization of all terms in the eddy-viscosity equation. The solution method described below is different from those found in [83] or [94]. The time advancement scheme can be written as

$$\frac{\tilde{\nu}^{n+1} - \tilde{\nu}^{n}}{\Delta t} = \frac{1}{2}\left(R^{n+1} + R^{n}\right), \tag{3.50}$$

where n is the time step index, Δt the time step size, and

$$\begin{aligned} R = & -\frac{1}{J}\frac{\delta}{\delta\xi_i}(q_i\tilde{\nu}) + c_{b1}\tilde{S}\tilde{\nu} - \frac{1}{Re}c_{w1}f_w\left(\frac{\tilde{\nu}}{d}\right)^2 \\ & + \frac{1}{\sigma Re}\left\{\frac{1}{J}\frac{\delta}{\delta\xi_i}\left[(1+\tilde{\nu})\alpha_{ij}\frac{\delta\tilde{\nu}}{\delta\xi_j}\right] + c_{b2}\frac{\alpha_{ij}}{J}\frac{\delta\tilde{\nu}}{\delta\xi_i}\frac{\delta\tilde{\nu}}{\delta\xi_j}\right\}. \end{aligned} \tag{3.51}$$

Rearranging equation (3.50) yields the nonlinear system of equations

$$F(\nu^{n+1}) \triangleq \nu^{n+1} - \nu^{n} + \frac{\Delta t}{2}(R^{n+1} + R^{n}) = 0, \tag{3.52}$$

which can be solved iteratively by the Newton method. In each Newton iteration the following linearized system of equation is solved:

$$\left[\frac{\partial F}{\partial \tilde{\nu}}\right]^{k-1} \delta\nu^{k} = -F^{k-1}, \tag{3.53}$$

where superscript k is the iteration index of Newton method, followed by solution update,

$$\tilde{\nu}^{k} = \tilde{\nu}^{k-1} + \delta\tilde{\nu}^{k}. \tag{3.54}$$

The operator $[\partial F/\partial\nu]$ in equation (3.53) is defined as

$$\begin{aligned} \left[\frac{\partial F}{\partial \nu}\right](\cdot) = (\cdot) + \frac{\Delta t}{2}\Bigg\{ & \frac{1}{J}\frac{\delta}{\delta\xi_i}\left[q_i(\cdot)\right] - c_{b1}\left(\tilde{\nu}\frac{\partial S}{\partial\tilde{\nu}} + S\right)(\cdot) \\ & + \frac{c_{w1}}{\sigma Re}\left[\left(\frac{\tilde{\nu}}{d}\right)^2\frac{\partial f_w}{\partial\tilde{\nu}} + 2f_w\frac{\tilde{\nu}}{d^2}\right](\cdot) \\ & + \frac{1}{\sigma ReJ}\frac{\delta}{\delta\xi_i}\left[\left(\alpha_{ij}\frac{\delta\tilde{\nu}}{\delta\xi_j}\right)(\cdot) + (1+\tilde{\nu})\alpha_{ij}\frac{\delta(\cdot)}{\delta\xi_j}\right] \\ & + \frac{c_{b2}\alpha_{ij}}{\sigma ReJ}\left[\frac{\delta(\cdot)}{\delta\xi_i}\frac{\delta\tilde{\nu}}{\delta\xi_j} + \frac{\delta\tilde{\nu}}{\delta\xi_i}\frac{\delta(\cdot)}{\delta\xi_j}\right]\Bigg\}. \end{aligned} \tag{3.55}$$

The iterations proceed until convergence, at which point the value of $\tilde{\nu}^k$ is taken to be $\tilde{\nu}^{n+1}$. The iterations are said to be converged when the criterion

$$\frac{|\delta\tilde{\nu}^{k}|}{|\tilde{\nu}^{k-1}|} < \epsilon \tag{3.56}$$

is satisfied, where $|\cdot|$ is the vector ∞-norm, and $\epsilon = 10^{-6}$ is used in all computations.

Directly inverting the left-hand-side operator in equation (3.53) in each Newton iteration is computationally expensive. Fortunately, since $[\partial F/\partial\tilde{\nu}]$ is used only to obtain $\delta\tilde{\nu}^k$ that drives the solution $\tilde{\nu}^k$ to approach the converged solution $\tilde{\nu}$, it can be approximated in a computationally efficient way provided that the iterations still converge fast enough. For this purpose, equation (3.53) is first rewritten as

$$\left[I + \frac{\Delta t}{2}\left(T_1 + T_2 + T_3 + D\right)\right]\delta\tilde{\nu}^k + \tilde{R}^k = -F^{k-1}, \tag{3.57}$$

where

$$\tilde{R} = \left[\frac{\partial F}{\partial \tilde{\nu}}\right]\delta\tilde{\nu} - \frac{\Delta t}{2}\left(T_1 + T_2 + T_3 + D_s\right)\delta\tilde{\nu},$$

and the operators T_1, T_2, T_3 and D_s are defined to be

$$T_1(\cdot) = \frac{1}{J}\frac{\delta q_1(\cdot)}{\delta\xi_1} - \frac{(1+c_{b2})\alpha_{11}}{J\sigma Re}\frac{\delta\tilde{\nu}}{\delta\xi_1}\frac{\delta(\cdot)}{\delta\xi_1} - \frac{1+\tilde{\nu}}{J\sigma Re}\frac{\delta}{\delta\xi_1}\left[\alpha_{11}\frac{\delta(\cdot)}{\delta\xi_1}\right], \tag{3.58}$$

$$T_1(\cdot) = \frac{1}{J}\frac{\delta q_2(\cdot)}{\delta\xi_1} - \frac{(1+c_{b2})\alpha_{22}}{J\sigma Re}\frac{\delta\tilde{\nu}}{\delta\xi_1}\frac{\delta(\cdot)}{\delta\xi_2} - \frac{1+\tilde{\nu}}{J\sigma Re}\frac{\delta}{\delta\xi_2}\left[\alpha_{22}\frac{\delta(\cdot)}{\delta\xi_2}\right], \tag{3.59}$$

$$T_3(\cdot) = \frac{\delta q_3(\cdot)}{\delta\xi_3} - \frac{(1+c_{b2})}{\sigma Re}\frac{\delta\tilde{\nu}}{\delta\xi_3}\frac{\delta(\cdot)}{\delta\xi_3} - \frac{1+\tilde{\nu}}{\sigma Re}\frac{\delta}{\delta\xi_3}\left[\frac{\delta(\cdot)}{\delta\xi_3}\right], \tag{3.60}$$

$$D_s(\cdot) = -c_{b1}S(\cdot) + \frac{2c_{w1}f_w\tilde{\nu}}{dRe}(\cdot). \tag{3.61}$$

By delaying the evaluation of $\tilde{R}$ by one iteration step, equation (3.57) becomes:

$$\left[I + \frac{\Delta t}{2}\left(T_1 + T_2 + T_3 + D_s\right)\right]\delta\tilde{\nu}^k = -\tilde{R}^{k-1} - F^{k-1}, \tag{3.62}$$

The left-hand-side operator in equation (3.62) can be approximately factorized to be

$$I + \frac{\Delta t}{2}\left(T_1 + T_2 + T_3 + D\right) \approx \left(I + \frac{\Delta t}{2}T_1\right)\left(I + \frac{\Delta t}{2}T_2\right)\left(I + \frac{\Delta t}{2}T_3\right)\left(I + \frac{\Delta t}{2}D_s\right). \tag{3.63}$$

To ensure that each of the right-hand-side operators in equation (3.63) are diagonally dominant, D_s is split into two parts:

$$D_s = D_s^+ + D_s^-, \tag{3.64}$$

where D_s^+ and D_s^- satisfy

$$D_s^+ = \begin{cases} D_s & \text{if } D_s \geq 0 \\ 0 & \text{if } D_s < 0 \end{cases}$$

and

$$D_s^- = \begin{cases} 0 & \text{if } D_s \geq 0 \\ D_s & \text{if } D_s < 0 \end{cases}$$

Using equations (3.63) and (3.64), the system of equations to be solved becomes

$$\left(I + \frac{\Delta t}{2}T_1\right)\left(I + \frac{\Delta t}{2}T_2\right)\left(I + \frac{\Delta t}{2}T_3\right)\left(I + \frac{\Delta t}{2}D_s^+\right)\delta\tilde{\nu}^k = -\tilde{R}^{k-1} - F^{k-1} - D_s^-\delta\tilde{\nu}^{k-1}. \tag{3.65}$$

Note that the evaluation of $\delta\tilde{\nu}$ associated with D_s^- has been delayed by one iteration step. Now equation (3.65) can be easily solved by inverting a series of diagonally-banded matrices.

Since Newton iterations converge quadratically, it takes only a few iterations to reach the convergence criterion (3.56). This is the case when the eddy viscosity at the previous time step is used as the initial guess for subsequent Newton iterations of the new time step. However, in some cases $\delta\tilde{\nu}$ changes abruptly between iterations to prevent $\tilde{\nu}$ from converging. To circumvent this, the following underrelaxation procedure is found effective to stabilize the calculation:

$$\tilde{\nu}^k = \tilde{\nu}^{k-1} + \gamma\delta\tilde{\nu}^k, \tag{3.66}$$

where $\gamma < 1$ is empirically determined. This is only needed early in the time advancement when a "non-physical" velocity field is given (such as a prescribed uniform flow past an object) and $\tilde{\nu} \approx 0$ is set as the initial condition for eddy viscosity. If the iterations are converging, $F \longrightarrow 0$ is recovered as $\delta\tilde{\nu} \longrightarrow 0$, regardless the use of underrelaxation procedure (3.66).

Due to numerical reasons, it is possible that on some grid points $\delta\tilde{\nu}^k < -\tilde{\nu}^{k-1}$ during the iterations. When this occurs, the local $\delta\tilde{\nu}$ value is replaced by $\max[\delta\tilde{\nu}^k, -\tilde{\nu}^{k-1}]$ to ensure that the solution $\tilde{\nu}^k$ is always positive. In all computations performed, this "clipping" procedure is only necessary on a small portion of grid points occasionally.

3.5 Parallelization

This section describes an implementation of the above algorithms on distributed-memory parallel computers, consisting of a number of processors, each of which has its own local memory. All processors are connected by high-speed networks. When a simulation is carried out using a subset of processors on parallel computers, data communication among the processors are accomplished by message passing. The MPI (Message Passing Interface) library, which has become widely available on many parallel computers, is used in the present computer code to coordinate data communication among individual processors. This approach permits explicit control of data communications and provides flexibility for code optimization. The resulting code requires little modification when porting to various parallel platforms.

The first step for parallelizing the algorithm is to divide the computational domain into a number of sub-domains, each of which is assigned to a processor. In the present implementation, the computational domain is divided into a number of sections along both the ξ_1 and ξ_3 directions, referred to as a "2-D" domain decomposition. The domain is divided along ξ_3 direction because it allows simulations to be carried out using a large spanwise domain size. The domain is divided along the ξ_1 direction because it allows simulations to be performed for problems that require a very large number of grid points along the streamwise direction (such as the transitional separated flow past an airfoil). Compared with the domain decomposition along only one direction (say, the spanwise direction), referred to as a "1-D" domain decomposition, used by our earlier implementation and by many others, the present method has the disadvantage of having additional inter-processor communication overhead. On the other hand, when the problem size is increased, the 1-D domain decomposition may suffer from speed and, more seriously, memory limitations, as each "slab" of the divided domain may still be too large for each individual processor to process. The present method scales well for very large problems and fundamentally eliminates such limitations. The communication overhead of the present method is likely to become less significant as more efficient networking hardware is developed.

The computational problem discretized by $N_1 \times N_2 \times N_3$ grid points is divided into N_{P1} and N_{P3} intervals along the ξ_1 and ξ_3 directions, respectively. The domain along the ξ_2 direction is

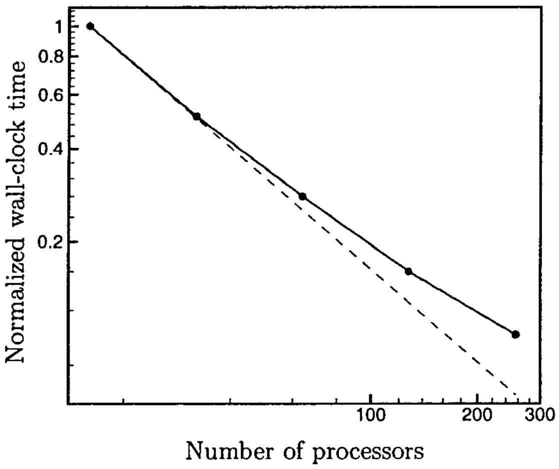

Figure 3.1: Parallel performance of the present simulation code. The dashed line is the ideal scalability, extrapolated based on the wall-clock time using 4 processors by assuming zero interprocessor communication time. The solid line is the actual scalability.

kept complete. This is advantageous when inverting banded matrices along the ξ_2 direction in the momentum and eddy-viscosity transport equations. Computing the right-hand-sides of discrete transport equations using explicit finite differences and the test filtering procedure in the dynamic model of LES on each processor are largely local operations, but require data communication among the processors storing flowfield information of adjacent subdomains. When solving the Poisson equation, the domain is divided in two ways. First, the domain is divided into segments of size $N_1/N_{P1} \times N_2/N_{P3} \times N_3$ to compute the FFTs along the ξ_3 direction. Second, the domain is re-arranged to have size $N_1/N_{P1} \times N_2 \times N_3/N_{P3}$ for the multigrid iterations, during which each processor communicates with the processors holding adjacent subdomains at each multigrid level. Some computations are overlapped with communication to improve parallel scalability. If the direct method is used for solving Poisson equation, the domain is divided into size $N_1/N_{P1} \times N_2/N_{P3} \times N_3$ to compute FFTs in ξ_3 direction and then into size $N_1 \times N_2/N_{P1} \times N_3/N_{P3}$ to compute FFTs along ξ_1 direction.

From a practical viewpoint, the most reliable way to check the parallel performance is to measure its absolute performance. We examine the overall scalability of the present parallel code by measuring the wall-clock execution time of simulations using different number of processors. The result is shown shown in figure 3.1 using up to 256 processors. Ideally the elapsed time would be inversely proportional to the number of processors. The curve eventually deviates from the ideal scalability curve due to increased ratio of communication time to computation time. The reduction in wall-clock computation time using multiple processors shows the advantage of parallelization.

Chapter 4

Simulations and Validations

In order to check the accuracy and efficiency of the parallel simulation methods described in Chapter 3, and to facilitate our study of separated flow control in subsequent chapters, a number of simulations are carried out in this chapter. The flows considered here are:

- Plane channel flow
  - DNS of 3D linear disturbance growth in a laminar base flow;
  - DNS of turbulent channel flow at Reynolds numbers $Re_\tau = 180$ and $Re_\tau = 590$;
  - DES and LES of turbulent channel flow at $Re_\tau = 2 \times 10^3$;
  - DES of turbulent channel flow at $Re_\tau = 2 \times 10^4$.
- Flow past a circular cylinder
  - DNS of cylinder flow at $Re_D = 300$;
  - DES of cylinder flow at $Re_D = 3900$.
- Flow past a NACA0012 airfoil
  - DNS at $Re_c = 1 \times 10^4$;
  - DES at $Re_c = 1 \times 10^5$.

The simulation of the growth of a linear disturbance in a channel is to check the spatial and temporal accuracy of the present solver. The DNS of turbulent channel flow and circular cylinder flow are to check the accuracy of the baseline solver for wall-bounded flows and external flows, respectively. The DES cases of channel and circular cylinder flows are performed to examine a number of issues associated with the DES approach. The vast database found in the literature, using either experimental or numerical methods, of these two canonical flows allows a number of comparisons to be made with the present results. Finally, 2D and 3D simulations of flows past an NACA0012 airfoil are performed at various angles of attack.

4.1 Plane Channel Flow

Flow in a plane channel is chosen as a validation for internal flows. In the following channel computations, we define (x, y, z) to be the coordinates in the streamwise, wall-normal and spanwise directions, and (u, v, w) to be the corresponding velocity components, respectively. We sometimes use (u, v, w) interchangeably with (u_1, u_2, u_3) for notational simplicity. The channel half-width is δ, and the kinematic viscosity is ν.

First, the growth rate of three-dimensional linear disturbances in a laminar channel flow is compared with that of linear stability theory [78]. Second, direct numerical simulations of fully-developed turbulent channel flow at $Re_\tau = 180$ and $Re_\tau = 590$ are compared with the benchmark DNS results of Kim *et al.* [51] and Moser *et al.* [67]. Third, LES and DES of turbulent channel flow at $Re_\tau = 590$ are compared with the DNS results in [67]. Fourth, DES results of channel flow at $Re_\tau = 2000$ and $Re_\tau = 20000$ are reported and discussed.

4.1.1 Growth of Linear Disturbances

When a three-dimensional small disturbance of the form

$$\begin{cases} u'(x,y,z,t) = u_o \exp(k_x x + k_z z - \omega t), \\ v'(x,y,z,t) = v_o \exp(k_x x + k_z z - \omega t), \\ w'(x,y,z,t) = w_o \exp(k_x x + k_z z - \omega t), \end{cases} \tag{4.1}$$

with x, y, and z normalized by the channel half-width δ, is introduced to the base flow

$$u = 1 - y^2, \quad v = 0, \quad w = 0, \tag{4.2}$$

in a plane channel, the evolution of the disturbance can be predicted by the solution of the Orr-Sommerfeld equation, written in terms of wall-normal velocity v and wall-normal vorticity $w_y = \partial u/\partial z - \partial w/\partial x$ [78]. After solving the corresponding eigenvalue problem, the real and imaginary parts of the eigenvalues give the growth/decay rate and the wave speed of the evolving disturbances.

We introduce a small three-dimensional disturbance into a laminar plane channel flow, and compare the growth rate of disturbance energy, defined as

$$E(t) = \frac{1}{2} \int_0^{L_x} \int_0^{L_z} \int_{-\delta}^{\delta} (u'^2 + v'^2 + w'^2)\, dy\, dz\, dx, \tag{4.3}$$

based on computed flowfields, with those predicted by linear theory to test the accuracy of the flow solver. The computational settings follow. The initial velocity field is

$$\begin{cases} u(x,y,z,t) = 1 - y^2 + \varepsilon u', \\ v(x,y,z,t) = \varepsilon v', \\ w(x,y,z,t) = \varepsilon w', \end{cases} \tag{4.4}$$

where u', v' and w' are taken to be the eigenvectors of the Orr-Sommerfeld equation for disturbance wavenumber $(k_x, k_z) = (1/\sqrt{2}, 1/\sqrt{2})$. The disturbance amplitude is $\varepsilon = 10^{-6}$. The Reynolds number, defined as $Re_c = u_c\delta/\nu$, is $7500\sqrt{2}$. Under these conditions, the solution of the Orr-Sommerfeld equation has one unstable mode and its disturbance energy growth rate is $E(t)/E(0) = e^{2\omega_i t}$ where $\omega_i = 1.5803837 \times 10^{-3}$.

On a staggered grid system, like the one used in our solution method discussed in Chapter 3, the kinetic energy cannot be defined unambiguously [66]. In the present calculation, the disturbance energy is defined to be the sum of $u'^2/2$, $v'^2/2$ and $w'^2/2$, evaluated at their own (staggered-grid) locations. Also, the eigenvectors $\hat{v}$ and $\hat{\omega}_y$ obtained by a spectral method [78] are mapped to the corresponding Cartesian components using the relations

$$\begin{aligned} u' &= \hat{u} \exp[i(k_x x + k_z z)]\ , \\ v' &= \hat{v} \exp[i(k_x x + k_z z)]\ , \\ w' &= \hat{w} \exp[i(k_x x + k_z z)]\ , \end{aligned}$$

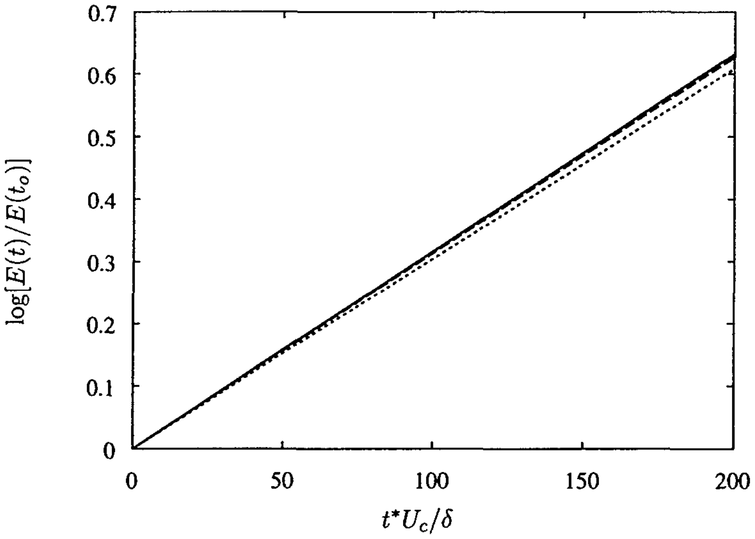

Figure 4.1: Effect of spatial resolution on perturbation energy growth rate at $Re = 7500\sqrt{2}$: —— linear stability theory; ········ $(N_x, N_y, N_z) = (64, 129, 64)$; ---- $(N_x, N_y, N_z) = (128, 129, 128)$.

where

$$\hat{u} = \frac{1}{k_x}\left(-k_z\hat{\omega}_y + i\frac{d\hat{v}}{dy}\right) ,$$

$$\hat{w} = \frac{i}{k_x^2 + k_z^2}\left(\hat{\omega}_y k_x + \frac{d\hat{v}}{dy}k_z\right) .$$

The eigenvectors generated by a spectral method correspond to a Chebyshev grid distribution, which is different from that used in the present finite-difference flow solver, in the wall-normal direction. A cubic spline interpolation technique is used to map the eigenvectors from the Chebyshev grids to the finite-difference grids.

The computational domain sizes in the streamwise, wall-normal and spanwise directions are $2\sqrt{2}\pi\delta$, 2δ, and $2\sqrt{2}\pi\delta$, respectively. In streamwise and spanwise directions the grids are uniformly distributed, while in wall-normal direction grids are compressed near the wall. Simulations are performed using two sets of spatial resolution, *i.e.*, $(N_x, N_y, N_z) = (64, 129, 64)$ and $(128, 129, 128)$, where N_x, N_y and N_z are number of mesh points in streamwise, wall-normal and spanwise directions, respectively, at constant CFL number 0.75. Figure 4.1 shows the effect of spatial resolution on the disturbance energy growth rate compared with that predicted by linear stability theory at $Re = 7500\sqrt{2}$. Using fourth-order central difference to discretize the convection terms, with $64 \times 129 \times 64$ mesh points, the error in energy growth rate from channel flow simulation is within 4% error from that of linear theory. Doubling mesh density in both x and z directions, resulting in a total of $128 \times 129 \times 128$ mesh points, reduces the error in energy growth rate to less than 1%, compared with that of linear theory. Using second-order central method for convection terms requires 50% more grid points in each direction to achieve similar accuracy. Both the fast transform method and the multigrid method to solve the Poisson equation for pressure produce essentially the same results for disturbance energy growth rate.

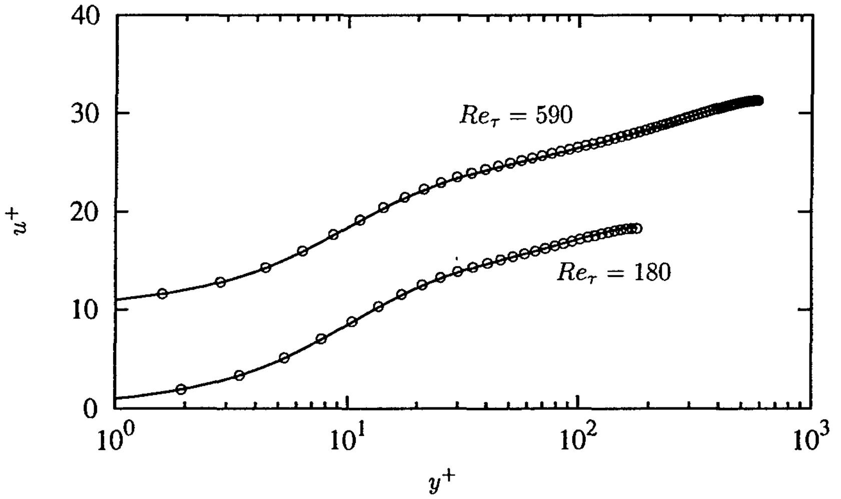

Figure 4.2: Mean streamwise velocity of turbulent channel flow at $Re_\tau = 180$ and $Re_\tau = 590$. —— present calculation; ○ spectral DNS [67].

4.1.2 Direct Numerical Simulation

The flow fields of fully developed turbulent channel flow are computed at Reynolds numbers $Re_\tau = 180$ and $Re_\tau = 590$, where the Reynolds number is defined to be $Re_\tau = u_\tau\delta/\nu$ based on the friction velocity $u_\tau = \sqrt{\tau_w/\rho}$. Periodic boundary conditions are applied in the spanwise and streamwise directions, and the no-slip boundary condition is applied at the walls. Grids are uniformly distributed along the streamwise and spanwise directions, and are compressed near the walls using a hyperbolic tangent function. A uniform pressure gradient, adjusted at each time step, is imposed along the streamwise direction to maintain constant mass flux across the channel throughout the simulation. The flowfields are well resolved by computational mesh; no turbulence model is used. The specific parameters used in these simulations are summarized in table 4.1.

Case	Re_τ	N_x	N_y	N_z	L_x/δ	L_y/δ	L_z/δ	Δx^+	Δz^+	Δy_w^+	Δy_c^+
A1	180	256	257	256	4π	2	$4\pi/3$	8.8	2.9	0.23	2.5
A2	590	512	397	512	2π	2	π	7.2	3.6	0.33	5.7

Table 4.1: Simulation parameters for channel flow DNS.

The initial flow field is the fully developed laminar plane channel flow,

$$u/u_c = 1 - (y/\delta)^2, \quad v = w = 0,$$

superposed by zero-mean random disturbances. The simulation is first advanced to reach a statistically steady state. Then the simulation continued for computing turbulence statistics. The standard Reynolds decomposition is used to split the mean velocity from the fluctuations:

$$\begin{aligned} u &= \overline{u} + u', \\ v &= \overline{v} + v', \\ w &= \overline{w} + w', \end{aligned}$$

where the mean velocity $(\overline{u}, \overline{v}, \overline{w})$ are computed by taking average in the streamwise and spanwise directions and in time. The fluctuations (u', v', w') are the deviations from the mean. All of the velocity components are normalized by the friction velocity u_τ in the following discussion. The mean streamwise velocity $\overline{u}^+$ are shown in Figure 4.2. The square root of normal Reynolds stresses $|u'^+|$, $|v'^+|$, $|w'^+|$ are shown in Figure 4.3. The Reynolds stresses $-\overline{u'^+v'^+}$ are shown in Figure 4.4. Good agreement between the current results and those reported in [67] is found at both Reynolds numbers.

4.1.3 Large-eddy and Detached-eddy Simulation

The large-eddy simulation (LES) and the detached-eddy simulation (DES) of turbulent channel flow are performed at $Re_\tau = 590$ to compare them with those from previous DNS results. In addition, two channel DES at $Re_\tau = 2000$ and $Re_\tau = 20000$ are carried out to explore the grid size effect in DES. The domain size of all cases is $(2\pi\delta, 2\delta, \pi\delta)$ in the streamwise, wall-normal, and spanwise direction, respectively. The main simulation parameters are summarized in Table 4.2.

Case	Re_τ	N_x	N_y	N_z	Δx^+	Δz^+	Δy_w^+	Δy_c^+	d_s^+	Type
B1	2000	64	129	32	196	196	0.3	89	136	DES
B2	2000	64	129	64	196	98	0.3	89	136	DES
B3	2000	128	129	64	98	98	0.3	89	65	DES
B4	20000	64	129	32	1963	1963	0.5	1507	1307	DES
B5	20000	64	129	64	1963	982	0.5	1507	1307	DES
B6	20000	128	129	64	982	982	0.5	1507	690	DES
B7	20000	128	129	128	982	491	0.5	1507	690	DES
C1	590	64	129	64	58	29	0.5	21		LES
C2	590	96	129	96	37	19	0.5	21		LES
C3	590	128	129	128	29	14	0.5	21		LES

Table 4.2: Simulation parameters of channel flow DES and LES.

In DES, the grid sizes along the streamwise and spanwise directions, Δx and Δz respectively, are constant, while the grids near the wall are compressed. The first grid point in the wall-normal direction is below $y^+ = 1$ in all cases. Unlike DNS and wall-resolving LES, DES can have a relatively large grid stretching ratio in the viscous sublayer and buffer layer. This usually results in a small number of grid points covering the region including the buffer layer and the bottom of the logarithmic layer, even for large Reynolds numbers. This works well because in this region the flow is computed using the RANS approach. While the flow in this region is unsteady, the fine-scale turbulent fluctuations seen in DNS and wall-resolving LES are largely absent. Grid expansion ratio becomes smaller in the bulk of logarithmic layer up to the channel centerline to provide appropriate resolution.

Following Nikitin *et al.* [69], in DES the distance function of the S-A model is defined as

$$d = \min[\tilde{d}, C_{DES} \max(\Delta x, \Delta y, \Delta z)] \ . \tag{4.5}$$

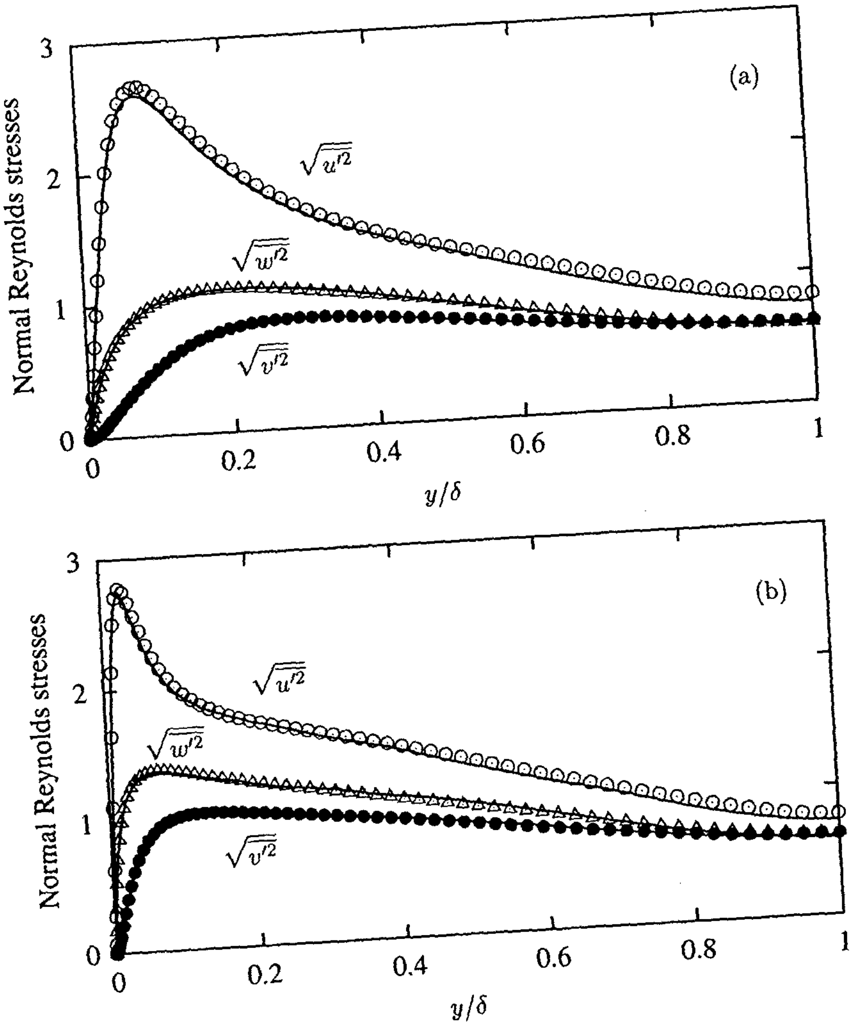

Figure 4.3: Normal Reynolds stresses of turbulent channel flow. (a) $Re_\tau = 180$. (b) $Re_\tau = 590$. —— present calculation; Symbols: spectral DNS [67] ○ u'^+ ● v'^+ △ w'^+ .

In the near-wall region, wall distance $\tilde{d}$ is small, so $d = \tilde{d}$; the DES operates in its "RANS mode" using the standard S-A turbulence model. Beyond a distance d_s from the wall, the DES switches to its "LES mode" by using the modified definition of d in the S-A model. In the channel flow DES, if Δy is smaller than $\max(\Delta x, \Delta z)$ across the channel, effectively $d = C_{DES}\max(\Delta x, \Delta z)$ is used across the entire LES region, and the grid size variation in the wall-normal direction is completely ignored. This is in contrast to the filter width in LES calculation, *i.e.*,

$$\Delta = (\Delta x \Delta y \Delta z)^{1/3} . \tag{4.6}$$

While other possibilities of defining Δ and the value of C_{DES} exist [74], in all results reported here we used the definition proposed in [69] for Δ and set $C_{DES} = 0.65$.

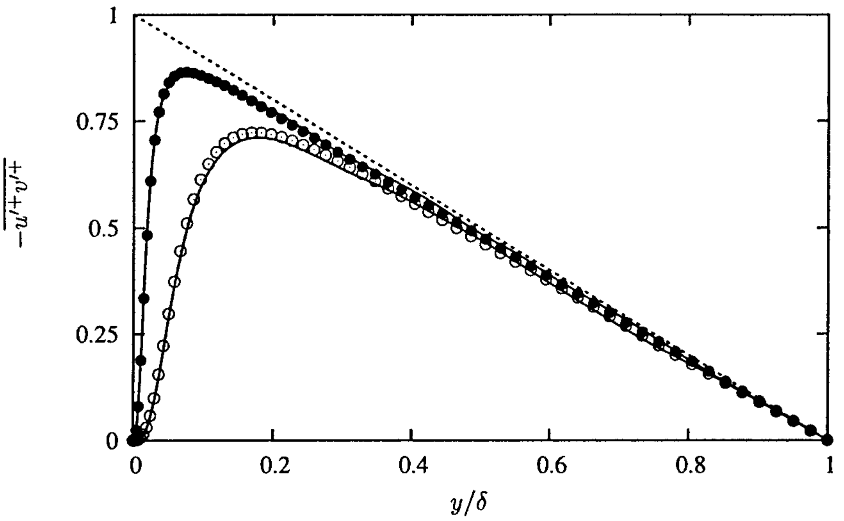

Figure 4.4: Reynolds stress $-\overline{u'^+v'^+}$ across the channel. —— present calculation; ○ spectral DNS at $Re_\tau = 180$, ● spectral DNS at $Re_\tau = 590$.

Figure 4.5 shows the mean streamwise velocity for $Re_\tau = 590$ using LES with three different grid sizes. The LES solution consistently over-predicts the mean streamwise velocity, but the difference between the LES solution and the "log-law" $u^+ = \log(y^+)/0.41 + 5.2$ becomes smaller when the mesh is systematically refined. The over-shooting appears to be a general trend when using a low-order finite difference scheme (like the present one), seen in the computations by other researchers (*e.g.*, [74]) as well.

The DNS, LES and DES fields for $Re_\tau = 590$ on a cross-flow (y-z) plane are compared in Figure 4.8. The near-wall small-scale turbulent motion, seen in the DNS and LES cases, is largely absent in DES. This is mainly because the small-scale motions are damped out by the high eddy viscosity level generated by the RANS model in this region.

Figure 4.6 compares the mean streamwise velocity for $Re_\tau = 2000$ from cases B1, B2 and B3. These cases show that the value of d_s plays an important role in DES. In cases B1 and B2, the solutions follow that of the log-law up to $y^+ \approx 100$, and has an up-shift to a "new" log-law, approximately parallel to the standard one. The improved accuracy of case B2 appears to be due to higher resolution in z direction. In case B3, the mean flow prediction is worse than cases B1 and B2. The streamwise grid size in case B3 is 0.1δ, which results in $d_s^+ = 65$ and causes the switch from RANS to LES region to take place close to the bottom of the logarithmic layer. Therefore the whole logarithmic layer is up-shifted. The low value of d_s results in higher value of $\partial\overline{u}^+/\partial y^+$, even though case B3 has better resolution in the streamwise and spanwise directions than cases B1 and B2.

Figure 4.7 shows the mean streamwise velocity profiles for $Re_\tau = 20000$ for cases B4, B5 and B6. Since the Reynolds number is sufficiently large, a logarithmic layer is clearly seen. Once the switch from RANS to LES takes place, an up-shift is again observed in the mean flow profile like the previous case at a lower Reynolds number, resulting in a log-law with higher intercept than the standard one.

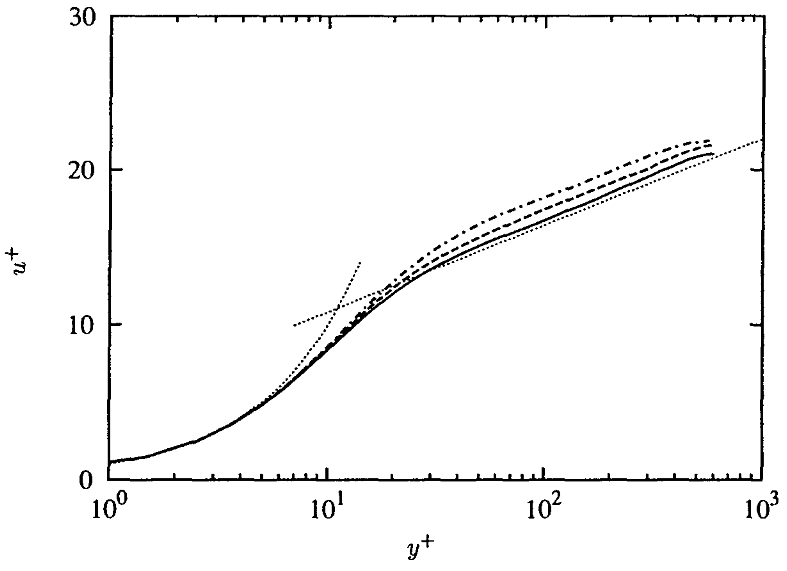

Figure 4.5: Mean streamwise velocity profiles of channel flow LES at $Re_\tau = 590$. —— case C1; ---- case C2; —-— case C3; ········ $u^+ = y^+$ and $u^+ = \log(y^+)/0.41 + 5.2$.

Case	Re_D	N_r	N_θ	N_z	Scheme	Type
A1	300	257	257	64	CDS	DNS
A2	3900	257	257	64	CDS	DES
A3	3900	257	257	64	Uw	DES
A4	3900	257	257	64	Uw	NM
A5	3900	257	257	64	Uw	URANS

Table 4.3: Simulation parameters for circular cylinder flows. CDS: 2nd-order central difference scheme; Uw: 3rd-order upwind scheme; NM: no model; URANS: unsteady RANS.

4.2 Flow Past a Circular Cylinder

Flow past a circular cylinder is chosen as a validation example for external flow. The patterns of cylinder flows are known to change significantly with Reynolds numbers, and has been a subject of intense study in the past. For subsequent discussions, we define the Reynolds number to be $Re_D = U_\infty D/\nu$, where U_∞ is the incoming free-stream velocity, D the cylinder diameter and ν the kinematic viscosity. For cylinder flow, DNS is performed at $Re_D = 300$, and DES is performed at $Re_D = 3900$. At $Re_D = 300$, the flow is unsteady, laminar and three-dimensional. At $Re_D = 3900$, the flow has a laminar separation and a turbulent wake.

For the $Re_D = 300$ case, the computational grid size on the cylinder surface is $0.002D$ and $0.015D$, along the wall-normal and spanwise directions, respectively. The surface grid size along the azimuthal direction varies from $0.010D$ to $0.015D$, clustered towards the downstream side of the cylinder to provide better resolution in the wake. Uniform mesh is used along the spanwise direction. The total number of grid points used is 256 × 257 × 64 along the azimuthal, radial

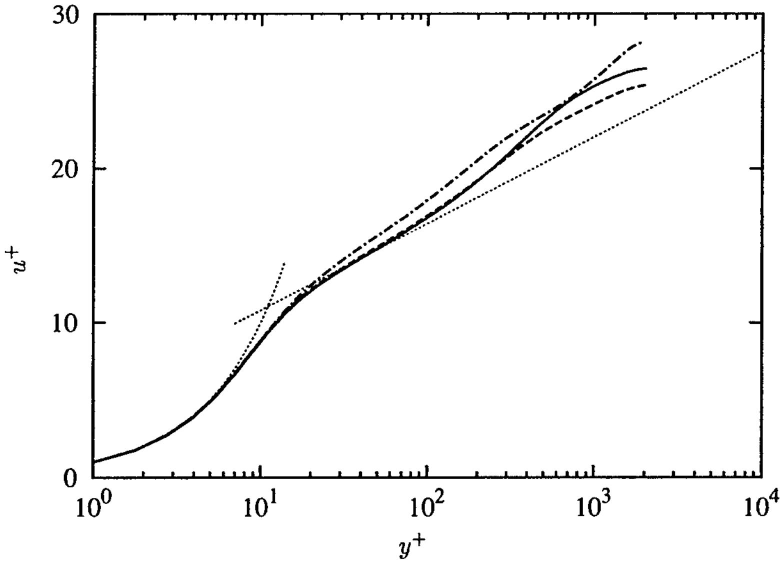

Figure 4.6: Mean streamwise velocity profiles of channel flow DES at $Re_\tau = 2000$. —— case B1; ---- case B2; —·— case B3; ········ $u^+ = y^+$ and $u^+ = \log(y^+)/0.41 + 5.2$.

and spanwise directions. The distance from the cylinder center to the computational boundary is approximately $22D$. The spanwise domain size is πD. The uniform flow

$$u = 1\,, \quad v = 0\,, \quad w = 0\,,$$

and the convective boundary condition

$$\begin{aligned} \frac{\partial u}{\partial t} + U_c \frac{\partial u}{\partial x} &= 0\,, \\ \frac{\partial v}{\partial t} + U_c \frac{\partial v}{\partial x} &= 0\,, \\ \frac{\partial w}{\partial t} + U_c \frac{\partial w}{\partial x} &= 0\,, \end{aligned}$$

are prescribed at the inflow and outflow boundaries, respectively, where U_c is the convection velocity at the outflow plane adjusted to maintain global mass conservation. The periodic boundary condition is used in the spanwise direction.

Uniform flow is prescribed as the initial condition. The simulations are advanced from $t = 0$ to $t = 200D/U_\infty$ to remove any effects from the initial condition. Then the solution is advanced for another $50D/U_\infty$ to compute statistics. The mean quantities are computed by averaging over the spanwise direction and in time. The results are compared with those from the spectral DNS by Mittal and Balanchandar [62]. The mean streamwise and vertical velocity profiles in the wake region $1.2 \leq x/D \leq 3$ are shown in Figure 4.9 and Figure 4.10 respectively. Generally good agreement is observed.

The computational settings for the $Re_D = 3900$ DES case follow. The surface grid sizes are $0.002D$, $0.015D$ in the wall-normal and spanwise directions, respectively, and varies from $0.01D$ to

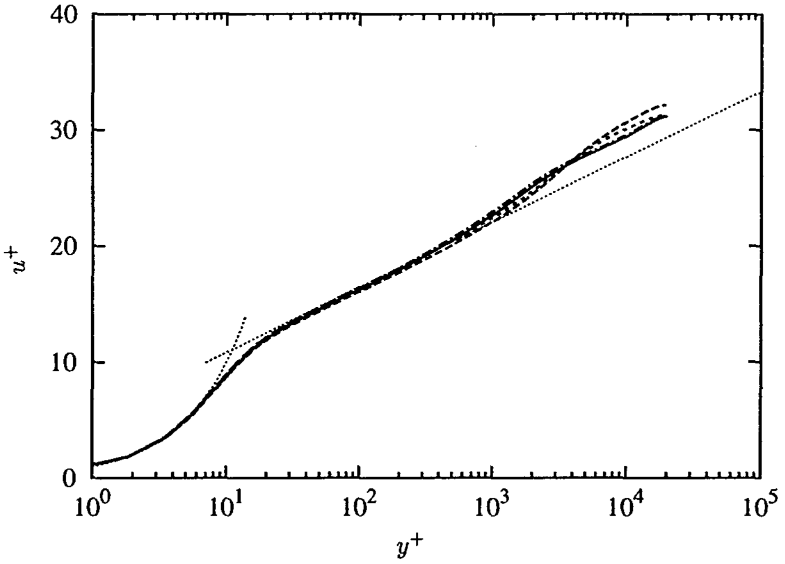

Figure 4.7: Mean streamwise velocity profiles of channel flow DES at $Re_\tau = 20000$. —— case B4; ---- case B5; —·— case B6; ········ $u^+ = y^+$ and $u^+ = \log(y^+)/0.41 + 5.2$.

$0.015D$ in the azimuthal direction, clustered toward the downstream side of the cylinder. The inflow and outflow boundary conditions for velocity components are similar to those of the $Re_D = 300$ DNS case. The eddy viscosity in the (laminar) freestream should be zero, but for numerical reasons, $\nu_t = 10^{-12}$ is used at the inflow plane. This value is several orders of magnitude smaller than the molecular viscosity (at $Re_D = 3900$), and is believed to have little impact on the results in the laminar region. At the outflow plane, a convective boundary condition is applied. Starting from uniform flow initial condition, the simulation advanced $80D/U_\infty$ units to reach a fully developed turbulent field. Statistics are computed over the next $80D/U_\infty$ time units, averaging in the spanwise direction and in time. The Strouhal number St and drag coefficient C_D are compared with those of the B-spline LES of Kravchenko & Moin, listed in table 4.4.

Case	St	C_D
2nd-order CDS	0.21	1.02
3rd-order upwind	0.2	0.9
ref [K&M]	0.21	1.01

Table 4.4: Computed global quantities of cylinder flow at $Re_D = 3900$

To examine the effects of spatial discretization schemes in DES, additional simulations are performed using third-order upwind difference schemes for the convection terms of the Navier-Stokes equations, with and without the DES model. The mean streamwise and vertical velocity profiles at $x/D = 1.06$, $x/D = 1.54$ and $x/D = 2.02$ are shown in Figure 4.11 and Figure 4.12, respectively. In general, good agreement is observed when using second-order central difference method. The results of 2D URANS (using Spalart-Allmaras turbulence model) simulation are also

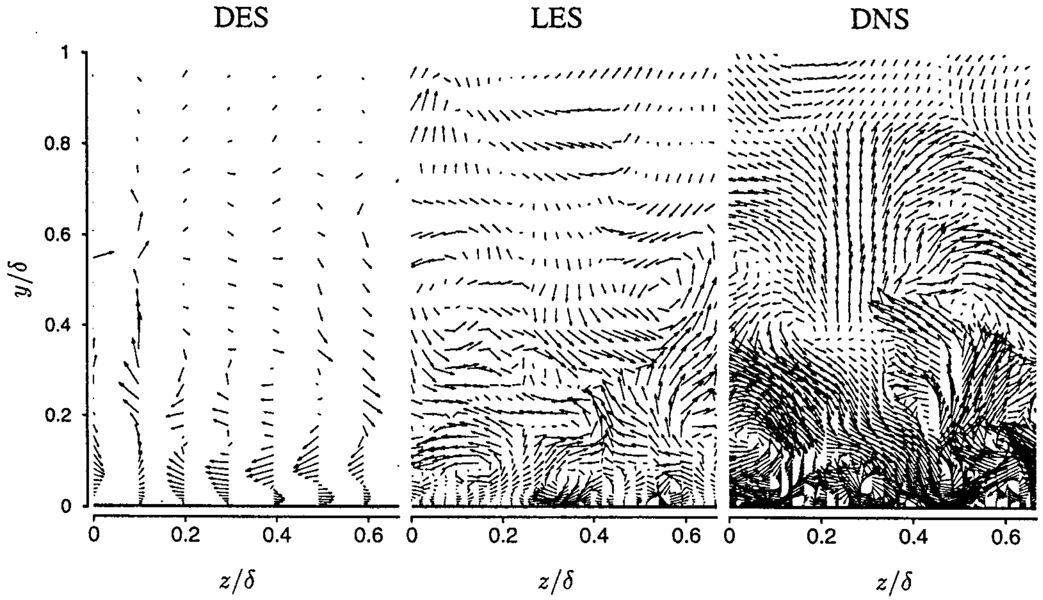

Figure 4.8: Comparison DNS, LES and DES fields at $Re_\tau = 590$ on a y-z plane.

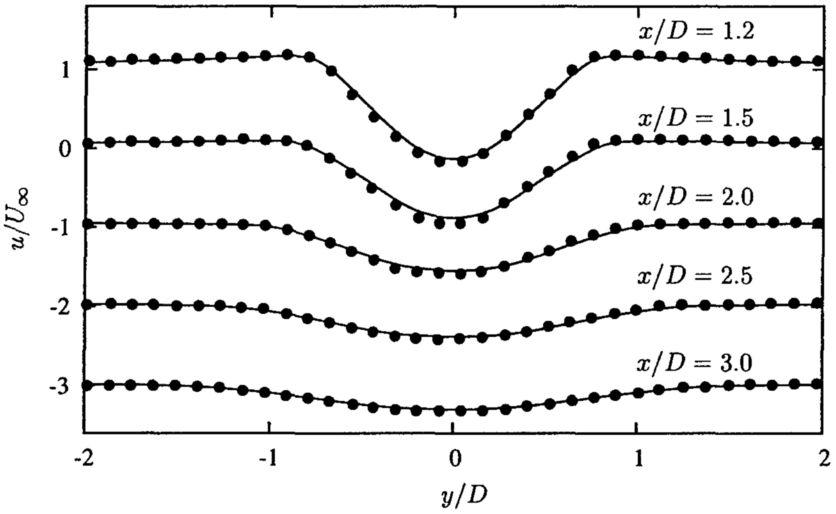

Figure 4.9: Mean streamwise velocity profiles downstream of a circular cylinder at $Re_D = 300$. —— present calculation; • Spectral DNS by Mittal & Balachandar.

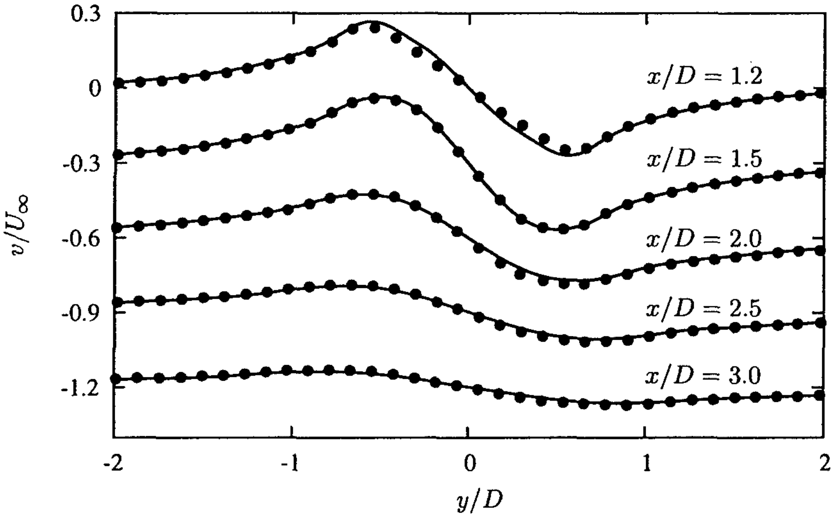

Figure 4.10: Mean vertical velocity profiles downstream of a circular cylinder at $Re_D = 300$. —— present calculation; • Spectral DNS of Mittal & Balachandar.

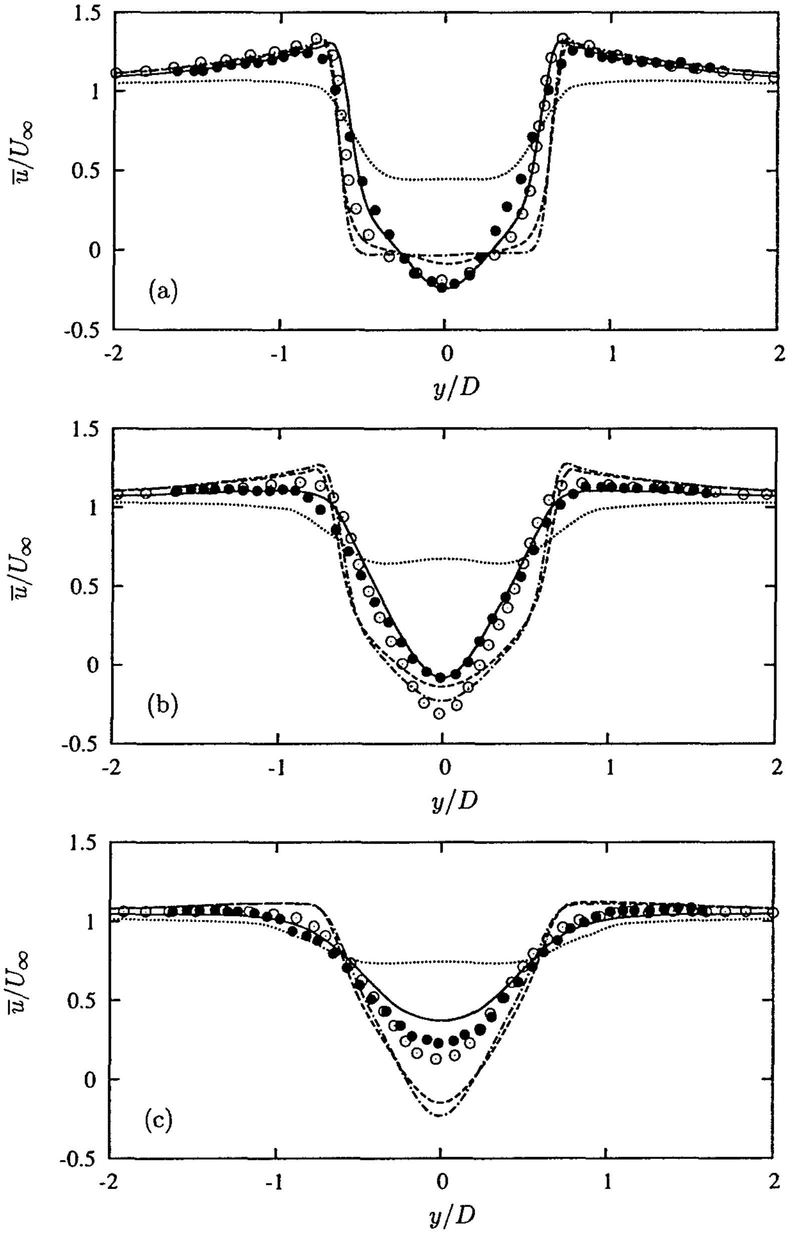

Figure 4.11: Mean streamwise velocity at $Re_D = 3900$. (a) $x/D = 1.06$; (b) $x/D = 1.54$; (c) $x/D = 2.02$. —— DES using second-order central difference scheme; ---- DES using 3rd-order upwind scheme; —·— No model using 3rd-order upwind scheme; ······· 2D unsteady RANS; ○ Kravchenko & Moin (B-spline LES); ● Lourenco & Shih (Experiment).

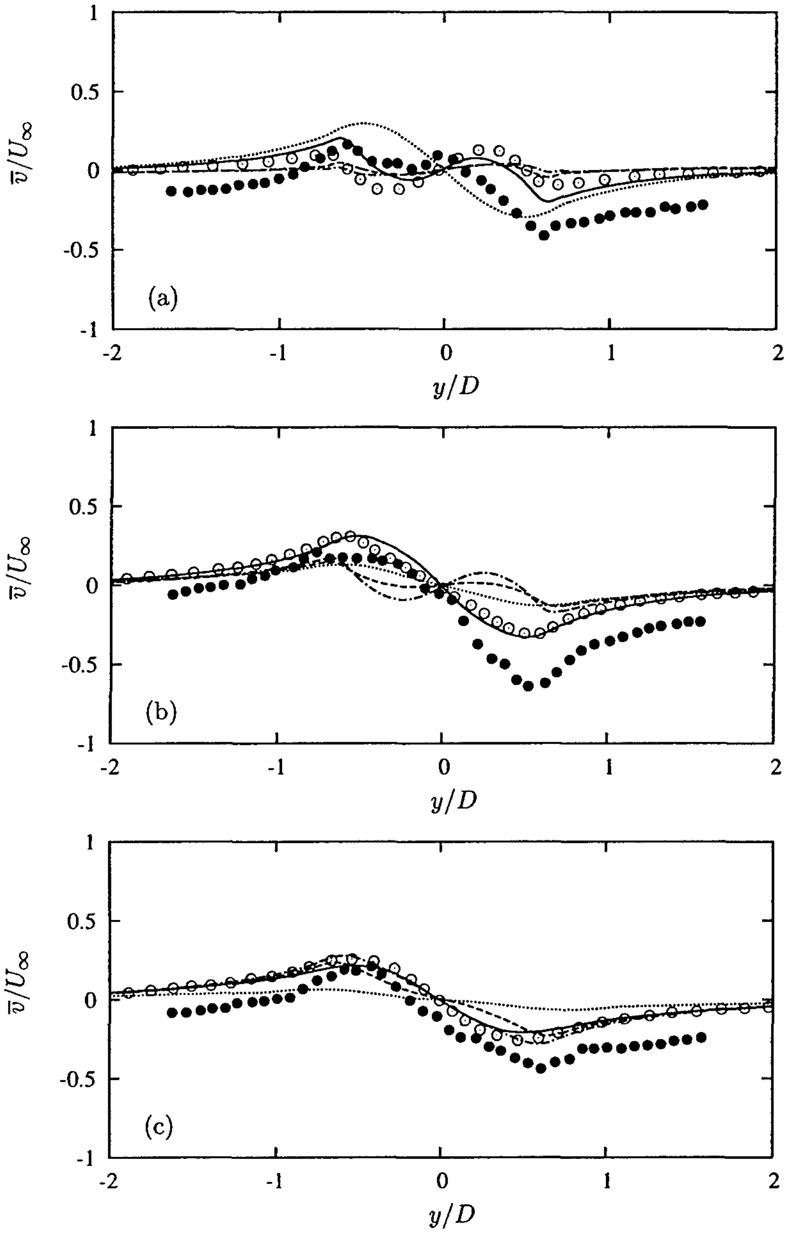

Figure 4.12: Mean vertical velocity at $Re_D = 3900$. (a) $x/D = 1.06$; (b) $x/D = 1.54$; (c) $x/D = 2.02$. —— DES using second-order central difference scheme; ---- DES using 3rd-order upwind scheme; —·— No model using 3rd-order upwind scheme; ········ 2D unsteady RANS; ○ Kravchenko & Moin (B-spline LES); • Lourenco & Shih (Experiment).

shown for comparison. Based on these results, it is clear that the non-dissipative second-order central difference scheme produces superior results than the dissipative third-order upwind scheme.

4.3 Flow Past an Airfoil

The NACA0012 airfoil is selected for all airfoil calculations. The Reynolds number $Re_c = U_\infty c/\nu$, where c is the airfoil chord length, U_∞ is the freestream velocity, and ν is the kinematic viscosity, ranges from 10^3 to 10^5. For low Reynolds number cases, the flow is assumed laminar and calculations are carried out without using turbulence models. DES is used for high Reynolds number turbulent cases. The key computational parameters of all cases performed are summarized in Table 4.5.

Case	Re_c	AOA	N_c	N_w	N_z	ν_T
A1	1×10^3	20°	257	129	1	0
A2	2×10^3	20°	257	129	1	0
A3	3×10^3	20°	257	129	1	0
A4	4×10^3	20°	257	193	1	0
A5	5×10^3	20°	257	193	1	0
B1	1×10^4	20°	257	193	1	0
B2	1×10^4	20°	257	193	64	0
C1a	1×10^5	15°	257	129	32	DES
C1b	1×10^5	15°	257	129	64	DES
C2a	1×10^5	20°	257	129	32	DES
C2b	1×10^5	20°	257	129	64	DES
C3a	1×10^5	25°	257	129	32	DES
C3b	1×10^5	25°	257	129	64	DES
C4a	1×10^5	30°	257	129	32	DES
C4b	1×10^5	30°	257	129	64	DES
D1	1×10^5	45°	129	129	1	RANS

Table 4.5: Summary of airfoil simulation cases. N_c: number of grid points along the circumferential direction. N_w: number of grid points in the wall-normal direction. N_z: number of grid points in the spanwise direction. AOA: angle of attack. The column of ν_T indicates how the eddy viscosity is modeled/calculated: 0 means the simulation is carried out without a turbulence model.

4.3.1 Three-dimensional Effects

At higher Reynolds numbers, the three-dimensional effects are strong. It is known that two-dimensional calculations cannot predict the lift and drag coefficients correctly. To demonstrate the effects of three-dimensionality, flow past a NACA0012 airfoil at 20° angle of attack is computed at $Re_c = 1 \times 10^4$. In case B1, 257×193 grid points are used in the x-y plane. In case B2, the same grid distribution is used in the x-y plane as in case B1, but has 64 planes in the z-direction. The spanwise domain size for case B2 is $4c$. Starting from uniform flow, the flow field is first advanced for 200 c/U_∞ time units for case B1 to remove effects from the initial condition. The flow field is then advanced for another 200 c/U_∞ time units to compute statistics.

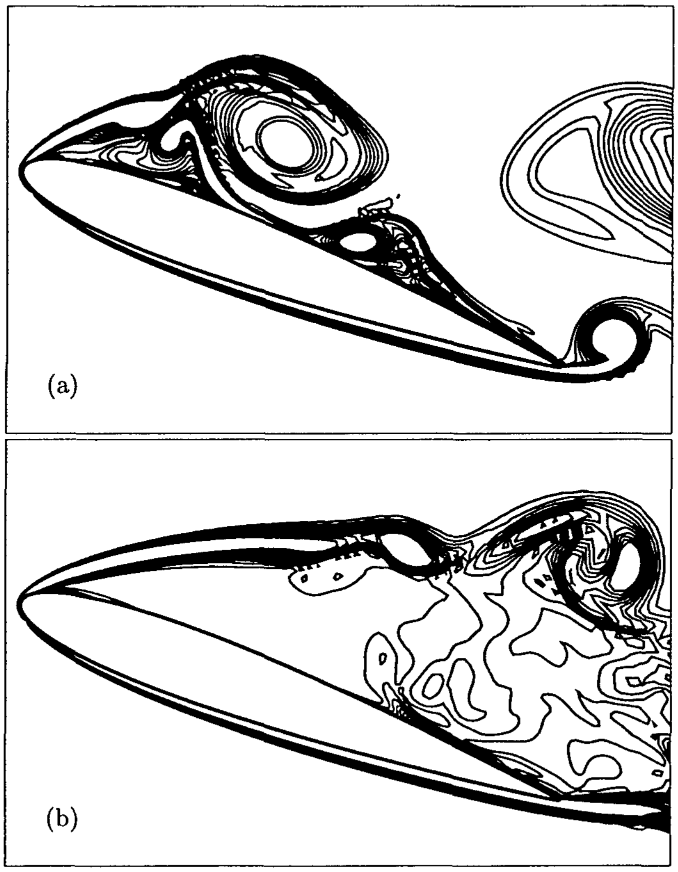

Figure 4.13: Total vorticity $|\omega|$ contours at $Re_c = 10000$. (a) case B1; (b) case B2.

The initial condition for case B2 is a two-dimensional flow field taken from a flow field of case B1, superimposed with zero-mean random disturbances. The flow field is then advanced for 100 c/U_∞ time units to remove the initial transients and to reach a fully three-dimensional flow field. The solution is then advanced for another 100 c/U_∞ time units to calculate statistics.

The contours of vorticity magnitude, defined as

$$|\omega| = \sqrt{\omega_x^2 + \omega_y^2 + \omega_z^2}\,, \tag{4.7}$$

of case B1 is shown in Figure 4.13(a). In this case, $|\omega| = |\omega_z|$, since the other two components are zero. The shear layer separates from the leading edge and develops a reverse flow region under it, forming a large vortex. The vortex induces a secondary vortex between itself and the trailing edge. These two vortices periodically shed from the suction side of the airfoil. These shedding vortices also interacts with the separated shear layer from the trailing edge. From the flow visualization of case B1, the process repeats but appears to be chaotic.

Figure 4.13(b) shows that total vorticity contours of case B2. The shear layer in this case is longer than that in case B1. Beyond the separation at the leading edge, the shear layer develops rollers, similar to those seen in canonical free-shear layers. The interaction with the wall is weaker

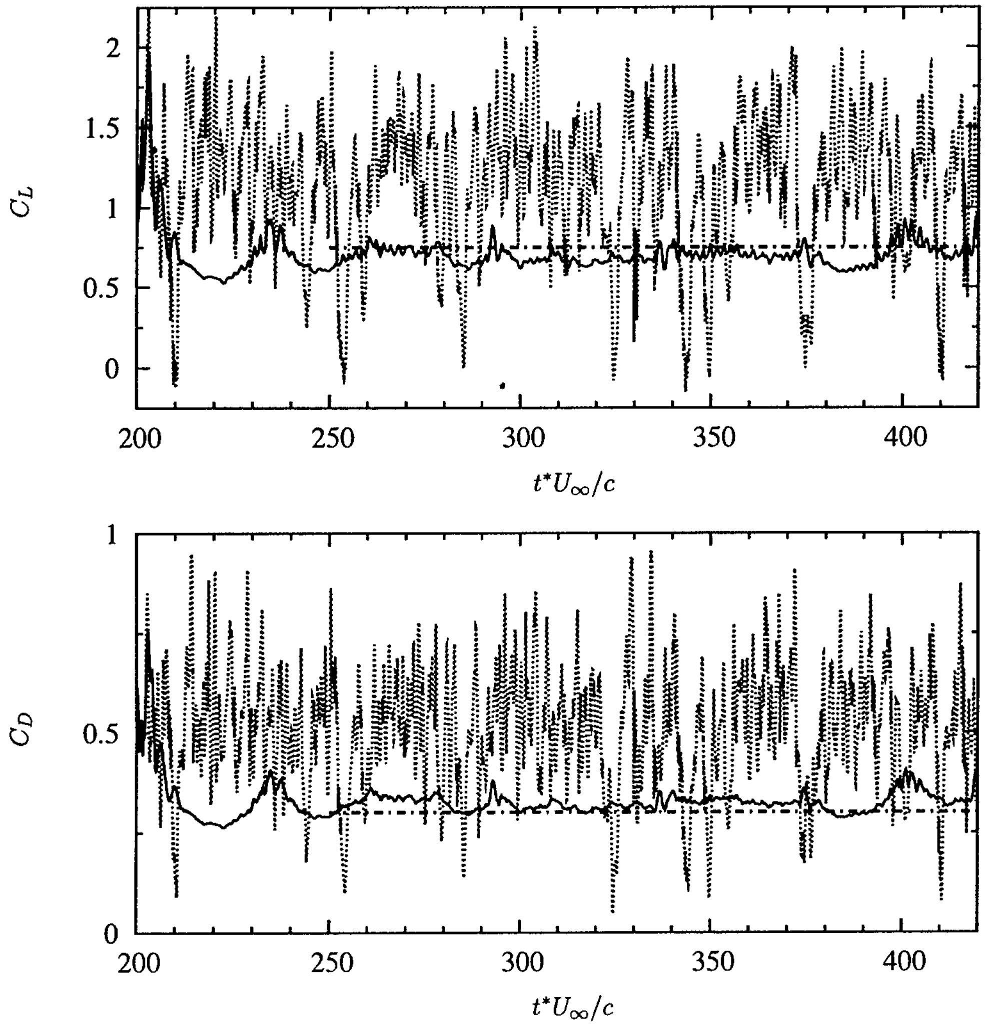

Figure 4.14: Evolution of lift and drag coefficients at $Re_c = 10000$. ········ case B1; —— case B2; —·— Experiment.

than that in case B1. The flow field becomes three-dimensional after the shear layer starts to roll up.

Time histories of drag coefficient C_D and lift coefficient C_L, defined as

$$C_D = \frac{2F_x}{\rho U_\infty^2} \,, \tag{4.8}$$

$$C_L = \frac{2F_y}{\rho U_\infty^2} \,, \tag{4.9}$$

where F_x and F_y are the total force (pressure force and viscous force added) along the streamwise x direction and the vertical y direction, respectively, of case B1 and case B2 are shown in Figure 4.14.

In case B2, the initial condition is the same as that of case B1 at $t = 200$. After adding spanwise disturbances, the flow field quickly became three-dimensional, but took approximately $50\,c/U_\infty$ for the lift and drag coefficients to level off, an indication that the flow field has reached a statistically steady state. The solution is then advanced to $100\,c/U_\infty$ to ensure removal of any effects from the initial condition.

It is seen that the level of lift and drag coefficients from two-dimensional (case B1) and three-dimensional (case B2) calculations differ by approximately 30%. The mean lift coefficient from case B1 is approximately 1.1, while that of case B2 is approximately 0.7, which is closer to the experimental value of 0.7. The mean drag coefficient of case B1 is approximately 0.45, while that in case B2 is 0.32, which is close to experimental value of 0.3. The fluctuation amplitudes of lift and drag coefficients from case B1 and B2 may not be compared directly, because they have been averaged along the spanwise direction for case B2 while no such averaging is done for case B1.

It is worth mentioning that the two-dimensional calculation of Hoarau *et al.* [36] is able to produce lift and drag coefficients very close to the corresponding experimental values at the same Reynolds number, while the present results show that only the three-dimensional calculation (case B2) is able to produce the correct levels of lift and drag coefficients. The cause of this discrepency is not clear at this time; more extensive grid independence tests are required in order to clarify the difference.

4.3.2 Detached-eddy Simulation

The flow past a NACA0012 airfoil at 20° angle of attack is computed using the DES approach. Two sets of grid points, which have 257×193×32 and 257×193×64 grid points along the circumferential, wall-normal and spanwise directions, respectively, are used. The computational domain size is approximately $10c$, c being the airfoil chord length, away from the airfoil surface for all cases. The spanwise domain size is c.

Before 3D DES is performed, the flow field of NACA 0012 at 45° angle of attack is calculated using the Spalart-Allmaras turbulence model (used in 2D URANS, unsteady Reynolds-averaged Navier-Stokes, mode). The computed spanwise vorticity, shown in Figure 4.15, is in qualitative agreement with that in Figure 3(d) of [82].

In order to discuss the results, a coordinate system must be first defined. Here we define the direction of each axis as follows. The direction of the streamwise coordinate axis x is aligned with the freestream velocity. The direction of the spanwise coordinate axis z is aligned with the span of the airfoil. The direction of the vertical axis y is normal to both x and z. Note that the vertical direction y is not normal to the chord of the airfoil based on this definition. The origin of this coordinate system is at the leading edge.

An overall view of this flow can be seen in Figure 4.16, where the isosurfaces of vorticity magnitude $|\omega| = \sqrt{\omega_x + \omega_y + \omega_z}$ is shown. The separated shear layer on the suction side of the airfoil is seen to have features similar to canonical unstable free-shear flows, in which small disturbances grow along the streamwise direction. The shear layer, under the influence of the flow in near-wall regions, eventually breaks up and forms a complicated flow pattern when merged with the boundary layer from the pressure side of the airfoil. Figure 4.17 shows plane views of instantaneous streamwise vorticity ω_x at streamwise locations $x/c = 0.2$, 0.4, 0.6, 0.8, 1.0 and 1.2. Note that according to the definition of the present coordinate system, locations corresponding to $x/c = 1.0$ and 1.2 are beyond the airfoil's trailing edge. It is seen that on the pressure side of the airfoil ω_x has negligible value along the entire surface, suggesting that the computed flow is two-dimensional and essentially steady there. On the suction side of the airfoil, the shear layer separates immediately after the flow passes the acceleration region around the leading edge. It can be seen that the

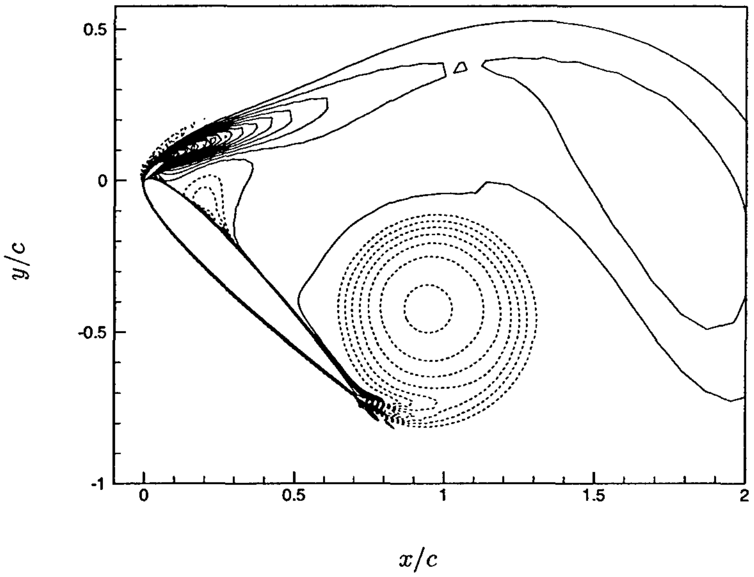

Figure 4.15: Instantaneous spanwise vorticity contours of an NACA0012 airfoil at 45° angle of attack from a two-dimensional RANS calculation at Re_c=100,000.

development of three-dimensionality initiates at near-wall regions and the separated shear layer then merge downstreams. Now we turn to the time-averaged view of this flow. The time-averaged velocity field of this flow is shown in Figures 4.18. It is seen that the separation shear layer does not reattach to the airfoil surface before it reaches the trailing edge. In order to discuss the computed time-averaged pressure coefficient C_p, defined as

$$C_p = \frac{p - p_\infty}{\frac{1}{2}\rho U_\infty^2}, \tag{4.10}$$

a different coordinate system is used, in which the x-axis is aligned with the chord of the airfoil, the z-axis is the spanwise direction, and the y-axis is normal to x and z. The time-averaged C_p computed using instantaneous fields is compared with that from the DES of [82] and that from the experimental data of [60], and is shown in Figure 4.19. In general good agreement is observed. The time-averaged pressure distributions using two different grids are essentially the same, suggesting that the computed mean flow is insensitive to spanwise resolution. The main discrepancy between the present time-averaged C_p and the experimental data in [60] is found near the leading edge on the suction side of the airfoil, where the pressure drop from [60] is higher than the present calculation and that of [82].

4.4 Summary

In this chapter, the efficiency and accuracy of the parallel computational code are tested for a number of representative flows. In general, good parallel scalability is achieved, even with very large problem sizes. For example, the wall-clock run time is approximately the same when the problem size (in terms of total number of grid points) and the number of processors are both

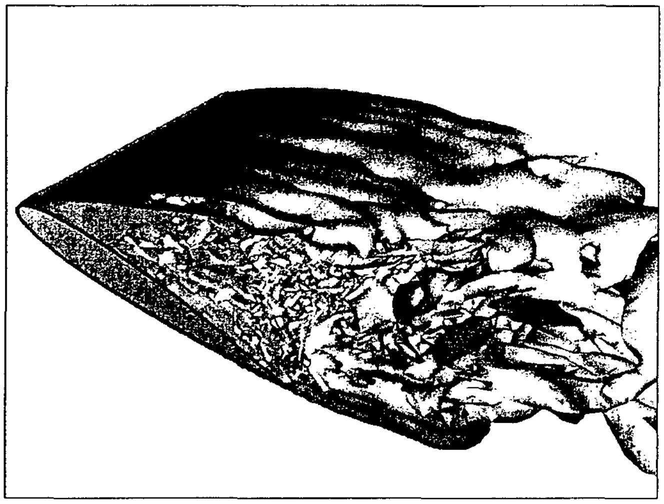

Figure 4.16: Isosurfaces of instantaneous vorticity magnitude $|\omega| = 10$ of the flow past a NACA0012 airfoil at 20° angle of attack and $Re_c = 100,000$ using DES.

doubled. This is possible mainly because of using the parallel algorithm discussed in Chapter 3. The accuracy level is the expected second-order in space and time. The computed turbulence statistics generally agree well with published results found in the literature. The use of generalized coordinates also allows many possibilities for future extension.

The results of turbulence statistics computed by the central difference method generally are better than those computed by the upwind-biased scheme for the channel and circular cylinder flows, in which grids are orthogonal or nearly orthogonal. However, in the airfoil calculations, significant dispersive errors are found due to the skewed mesh when the second-order central difference scheme is used. Although the upwind-biased scheme excessively damps small-scale turbulence, it gives more stable solutions. It is found necessary to use upwind-biased scheme to discretize the eddy viscosity transport equation to avoid oscillatory solutions of eddy viscosity.

Of particular interest is the evaluation of the DES approach as a tool for studying separated flows and their control. It turns out that, while the DES approach has the potential of treating high-Reynolds number flows, rigorous treatment of the RANS/LES interface remains an open question. In the current DES formulation [69] using the framework of the Spalart-Allmaras model, the switch between RANS and LES is predetermined by mesh (see equation (2.31)) but not by flow physics. This is a convenient feature due to its numerical robustness, observed in all of our test cases, and the fact that a single turbulence model equation is used throughout the domain.

However, the current RANS/LES interface treatment is not satisfactory, as it generates an artificial layer in which the solution transitions from RANS to LES. For the LES region near the interface, the eddy viscosity produced from the RANS side tends to be large, since in the RANS approach the eddy viscosity represents the effect of total Reynolds stresses. On the other hand, the eddy viscosity in the LES calculation represents the effects of the (unresolved) subgrid scale stresses only. As a result, the velocity fluctuations in the RANS region tend to be too small for the LES side. The overall effect of this is that the velocity fluctuations in the LES side are damped by the nearby RANS solution, and the RANS solution is driven by the nearby (resolved) fluctuations in LES solution. Thus, while the current DES approach represents a feasible way to compute high Reynolds number flows, the LES/RANS interface treatment should be improved in order to improve the accuracy of DES.

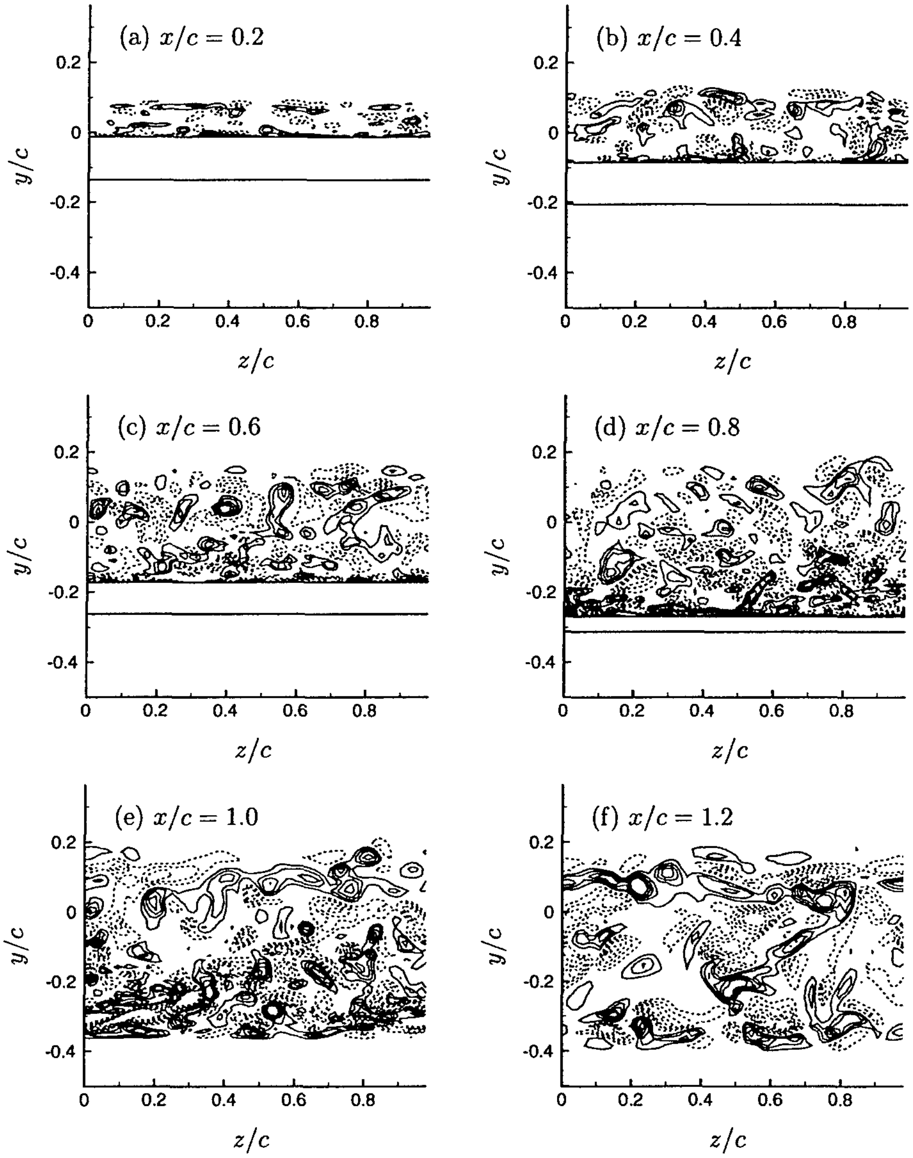

Figure 4.17: Contours of streamwise vorticity -10 to 10 with increments of 2 at various streamwise locations: $x/c = 0.2$, 0.4, 0.6, 0.8, 1.0 and 1.2 on y–z planes. The horizontal solid lines in (a)–(d) indicate the upper and lower surfaces of the airfoil. Mean flow is into the paper. Negative contours are dashed

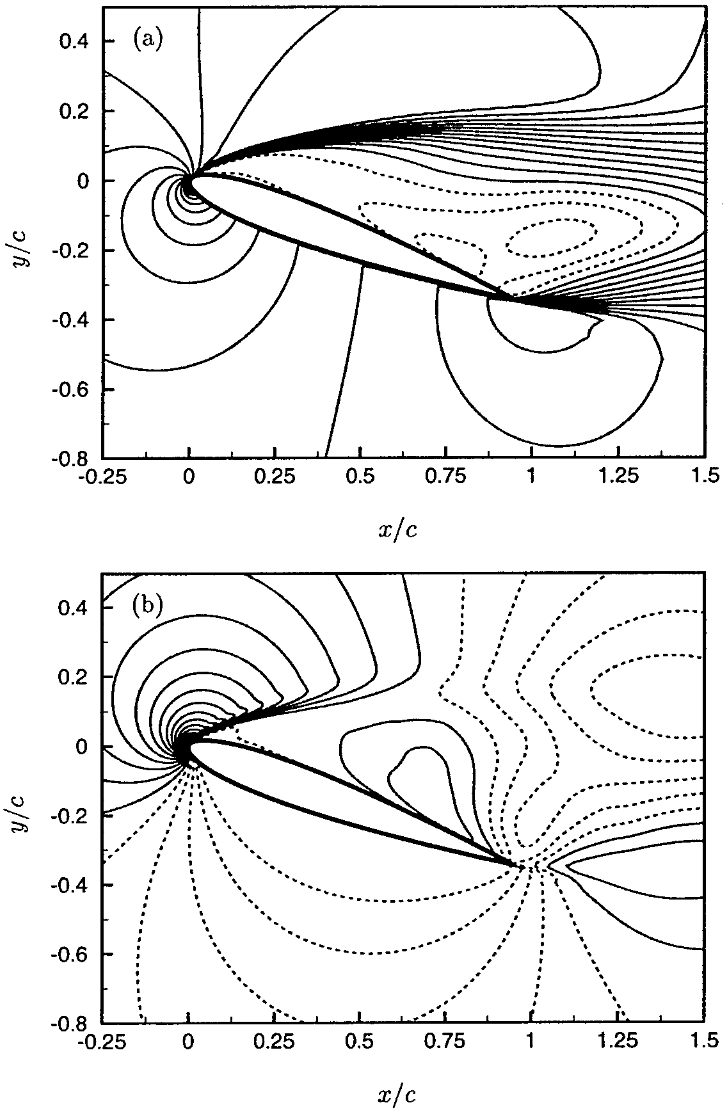

Figure 4.18: Time-averaged velocity field of a NACA 0012 airfoil at 20° angle of attack: (a) streamwise x-component, contour levels -0.2 to 1.2 with increment of 0.1; (b) vertical y-component, contour levels -0.2 to 1.2 with increment of 0.05. Negative contours are dashed.

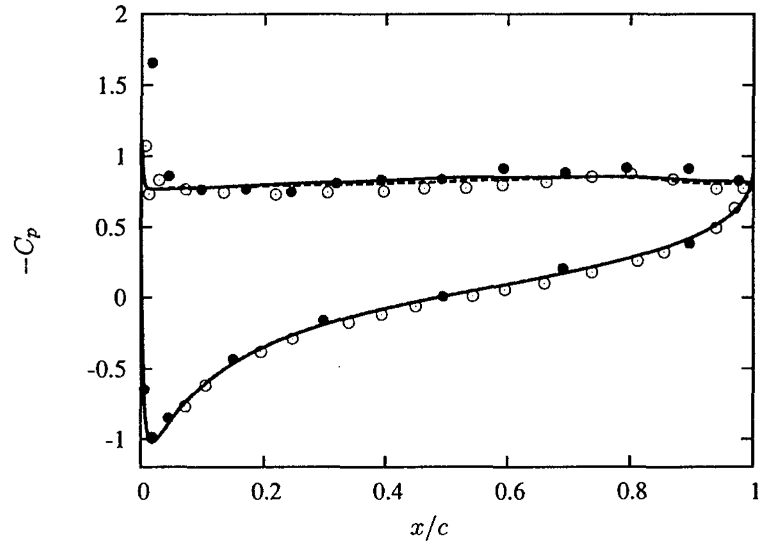

Figure 4.19: Time-averaged pressure coefficient C_p distribution on the surface of a NACA 0012 airfoil at 20° angle of attack: —— 257×193×32 grid points; ---- 257×193×64 grid points; ○ from [82]; ● from [60].

Chapter 5

Control-theoretic Approach

This chapter discusses the control-theoretic approach to be used for controlling separated flows in the present study.

5.1 Preliminaries

Linear optimal control theory is used in the present study to develop feedback control laws for separated flows. Applications of linear optimal control theory to flow control problems can be found in the literature, such as channel flow control [38, 56] and flat-plate boundary layer control [39]. In these applications, the derivation of a linear model of the flow system starts with linearizing the Navier-Stokes equations about a mean flow. The linearized equations are then numerically discretized, which can be viewed as approximating an infinite-dimensional system by a finite-dimensional system, to obtain a linear model for control synthesis. The linear system model is cast into the standard continuous-time finite-dimensional time-invariant state-space form,

$$\begin{cases} \dfrac{dx}{dt} = Ax + Bu, \\ \quad y = Cx + Du, \end{cases} \tag{5.1}$$

where x is the vector of state variables, u is the control (input) vector, y is the output vector, and (A, B, C, D) are the state-space system matrices [24]. Specifically, in these flow control studies using linear optimal control theory (*e.g.,* [10, 38, 56]), the linearized Navier-Stokes equations are written in a special form so that the state vector contains the wall-normal velocity and wall-normal vorticity. Consequently, the system matrix A is related to the Orr-Sommerfeld and Squires operators in shear flow stability theory [78]. In addition, Fourier decomposition is used along the homogeneous flow directions, so that the governing equations of the linear system are transformed into a series of decoupled ones corresponding to each wavenumber pair. Feedback control laws are computed for selected wavenumber components using linear optimal control theory (in the wavenumber space). This linear control approach works well, successfully achieving drag reduction in turbulent flows [38, 56]. The issues of using linear control in nonlinear turbulent flows have been discussed in [48, 49].

For complex flows, such as separated flow past an airfoil, however, the explicit representations of system matrices (A, B, C, D) in equation (5.1) are not readily available. In addition, standard Fourier decomposition is not available due to flow inhomogeneity and geometry, and the resulting system matrices are too large to handle. In the present study, the state-space models are estimated using certain input and output data sequences. Feedback control laws are then computed, based on the approximate model, using linear optimal control theory. The actuation is velocity blowing and suction at predefined locations on the wall and all measurement locations are also confined on the wall.

5.2 Linear Model

The first step in applying the linear optimal control theory to flow control problems is to obtain a linear system model. Specifically, our goal is to obtain a discrete-time linear finite-dimensional time-invariant state-space model,

$$\begin{cases} x(t+1) = Ax(t) + Bu(t), \\ \quad\; y(t) = Cx(t) + Du(t), \end{cases} \tag{5.2}$$

in which t is the time step index, and x, y and u are the state vector, output vector and control input vector, respectively, defined analogously to their continuous-time counterparts in equation (5.1). The physical time interval between t and $t+1$ is called the sampling time.

The use of a discrete-time system model is in contrast to previous works in flow control using linear control theory [10,38,56], where a continuous-time framework has been used. A continuous-time framework is a natural choice when the system model is derived directly from partial differential equations containing time derivative terms, *i.e.*, the linearized Navier-Stokes equations [10]. In the present study, since the system model is estimated from sampled input-output data sequences, a discrete-time framework appear to be more convenient. The conversion from discrete-time system models to continuous-time system models (and vice versa) is possible, but we use a discrete-time approach throughout this study.

Strictly speaking, since the flow dynamics is nonlinear, a nonlinear state-space model of the form

$$\begin{cases} x(t+1) = h(x(t), u(t)), \\ \quad\; y(t) = g(x(t), u(t)), \end{cases} \tag{5.3}$$

should be used, where g and h are nonlinear functions. It is known that using a nonlinear model will result in a much more difficult control problem for computing the feedback laws [1,58]. For the purpose of flow control, our goal is not to construct an accurate system model, but to construct a (preferably simple) system model, which can lead to effective feedback control laws. In the present study, only linear state-space models of the form (5.2) are considered. The following section describes the methods used to estimate state-space matrices in equation (5.2) using sampled input and output data sequences.

5.3 System Identification

In order to build a system model directly from sampled input-output data sequences, a model structure, which contains a number of model parameters, is first assumed, and then the model parameters are determined by minimizing the error norm (in a certain sense) between the model output and system output given the same input data sequence. Once the linear model is obtained, its equivalent state-space form (5.2) can be obtained and used for control synthesis.

Since the identified input-output model only represents an input-output equivalence to the original flow system, it corresponds to infinitely many state-space system models, all of which are related by a state-space similarity transformation. As a consequence, the state variables in the identified system do not have obvious physical meanings. This is in contrast to previous studies, where the state variables have well-defined physical meanings (*e.g.*, wall-normal velocity and wall-normal vorticity in [10, 56]). Here the time evolution of the state variables only represent the dynamics of the linear model in a certain set of state-space coordinates. On the other hand, both the system output and control input have well-defined physical meanings.

The system model constructed based on input-output data represents only the observable part of the system. The unobservable part of the system cannot be identified by the input-output data, and will not enter the system model. Therefore, it is important to place sensors and actuators (in the parameter estimation phase) at appropriate locations in order to make important dynamics of the system observable so that useful information can enter the linear model. In practice, this involves trial-and-error, because the optimal locations of sensors and actuators are not known in advance.

The convergence of the model parameters also requires attention. If the system is linear, and the (assumed) linear model contains sufficient degrees of freedom, then the model may converge to the true linear system when the number of data samples is sufficiently large. In contrast, if the system is nonlinear, the assumed linear model will not converge to it, since the nonlinear dynamics cannot be captured by a linear model no matter how many data samples are used. Nevertheless, some important features of the nonlinear system may still be captured by the linear model. In general, it not known in advance whether the dynamics captured by a linear model can result in a good feedback control, or the unmodeled nonlinear dynamics will significantly limit the performance of linear control. In the present study, only linear time-invariant system models are considered.

In the following sections, two identification methods that are used for separated flows are described. The identified linear system models are then used for control synthesis. Detailed information about the derivation of these methods can be found in the literature, *e.g.*, [57] and [70].

5.3.1 ARX Model

In the present study, the ARX (Auto-Regression with Exogenous Input) model [57] is considered to represent the input-output relations of the flow system. For a linear system with n_u input channels and n_y output channels, the ARX model can be written as

$$y(t) + \sum_{i=1}^{N} A_i y(t-i) = \sum_{i=0}^{N} B_i u(t-i), \tag{5.4}$$

where matrices A_i and B_i contain model constants to be determined, $y(t)$ is the output vector of length n_y, $u(t)$ is the input vector of length n_u, N is the model order. The matrix coefficients A_i $(1 \leq i \leq N)$, is a $n_y \times n_y$ matrix, and B_i $(0 \leq i \leq N)$ is a $n_y \times n_u$ matrix. Equivalently, the model can be expressed in a compact form,

$$y(t) = D(t-1)\theta, \tag{5.5}$$

where $D(t-1)$ is a linear function of input-output data,

$$y(t-1), y(t-2), \cdots, y(t-N), u(t-1), u(t-2), \cdots, u(t-N), \tag{5.6}$$

and θ contains the unknown model parameters, *i.e.*, the rows of A_i and B_i. To fit the model to input-output data, one seeks the best A_i and B_i that can minimize the difference between the system output and the model output using available sampled data for the same input sequence. One way of finding the best A_i and B_i is to solving the least square problem,

$$L\theta = R, \tag{5.7}$$

where

$$L = \begin{bmatrix} L_y & L_u \end{bmatrix}, \tag{5.8}$$

and

$$L_y = \begin{bmatrix} y^T(t_f-1) & y^T(t_f-2) & \cdots & y^T(t_f-N) \\ y^T(t_f-2) & y^T(t_f-3) & \cdots & y^T(t_f-N-1) \\ \vdots & \vdots & \vdots & \vdots \\ y^T(N) & y^T(N) & \cdots & y^T(1) \end{bmatrix}, \tag{5.9}$$

$$L_u = \begin{bmatrix} -u^T(t_f) & -u^T(t_f-1) & \cdots & -u^T(t_f-N) \\ -u^T(t_f-1) & -u^T(t_f-2) & \cdots & -u^T(t_f-N-1) \\ \vdots & \vdots & \vdots & \vdots \\ -u^T(N+1) & -u^T(N) & \cdots & -u^T(1) \end{bmatrix}, \tag{5.10}$$

where t_f is the number of samples and superscript T denotes the matrix transpose operation. The dimension of matrix L in equation (5.7) is $(t_f - N) \times (Nn_y^2 + Nn_yn_u + n_yn_u)$.

In practice, since the order of the system model to be identified is not known in advance, it is necessary to over-parametrize the system model by choosing a high system order to make sure that the model's degree of freedom is sufficiently high. When the matrix L has full rank, an efficient way to solve the least square problem is the QR method [31]. However, when the selected system order N is large, it is possible that L becomes rank-deficient as some of the columns of L are (numerically) nearly linearly dependent. When L is rank-deficient, there are infinitely many solutions of θ satisfying the least square problem, because the null space of L contains non-zero elements. Each of these infinitely many solutions is actually the sum of a minimum-norm solution, which lies in the row space of L, and an arbitrary component that lies in the null space of L. In the present study, the (unique) minimum-norm solution θ of the least square problem is used for parameter estimation. The minimum-norm solution can be written symbolically as

$$\theta = L^+R, \tag{5.11}$$

where L^+ is the pseudo-inverse of L, which can be computed by a method based on the singular value decomposition of L [31].

Once the model coefficient matrices A_i and B_i in equations (5.4) are found, the system model can be converted to an equivalent standard state-space form (5.2). In the present study, we chose to use the observer form [57], in which the system matrices (A, B, C, D) are related to model coefficients A_i and B_i by:

$$A = \begin{bmatrix} -A_1 & I & 0 & 0 & \cdots & 0 \\ -A_2 & 0 & I & 0 & \cdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & I \\ -A_N & \vdots & \cdots & \cdots & \cdots & 0 \end{bmatrix}, \quad B = \begin{bmatrix} B_1 - A_1B_0 \\ B_2 - A_2B_0 \\ \vdots \\ \vdots \\ \vdots \\ B_N - A_NB_0 \end{bmatrix}, \tag{5.12}$$

$$C = \begin{bmatrix} I & 0 & 0\cdots \end{bmatrix}, \quad D = B_0 \tag{5.13}$$

The state-space realization of the identified ARX model has n_yN states.

Over-parameterizing the system may introduce some states that are uncontrollable or unobservable (or nearly so). These states should be removed before control synthesis. This can be done by using the balanced truncation model reduction technique [32]. This model reduction technique starts with finding the similarity transformation matrix T such that the controllability Gramian W_p and observability Gramian W_o, defined as

$$W_p = \sum_{t=0}^{\infty} (A^*)^t C^* C A^t, \tag{5.14}$$

$$W_o = \sum_{t=0}^{\infty} A^t B B^* (A^*)^t, \tag{5.15}$$

in the infinite time interval become equal and diagonal in the new state coordinates. The state vector $\tilde{x}$ in the transformed space is related to the original state vector x by $\tilde{x} = Tx$. The state-space system in the new coordinates can be expressed as,

$$\tilde{x}(t+1) = \tilde{A}\tilde{x}(t) + \tilde{B}u(t), \tag{5.16}$$

$$y(t) = \tilde{C}\tilde{x}(t) + \tilde{D}u(t), \tag{5.17}$$

where

$$\tilde{A} = T^{-1}AT, \quad \tilde{B} = T^{-1}B, \quad \tilde{C} = CT, \quad \tilde{D} = D. \tag{5.18}$$

In equations (5.14) and (5.15) the Gramians W_o and W_p can be found by solving the Lyapunov equations,

$$A^* W_o A - W_o + C^* C = 0, \tag{5.19}$$

$$A W_p A^* - W_p + B B^* = 0, \tag{5.20}$$

using a method proposed by Barraud [7] based on Schur factorization of A. It is known that when A is stable, the solution to each of these Lyapunov equations is unique. It can be shown (*e.g.*, [32]) that the similarity transformation matrix T can be constructed by

$$T = F^{-*} U \Sigma^{1/2}, \tag{5.21}$$

where F is a factorization of W_o satisfying $W_o = FF^*$, and U and Σ are from the singular value decomposition,

$$F^* W_p F = U \Sigma V^*. \tag{5.22}$$

Next, the singular values of the Hankel matrix H, defined as

$$H = \begin{bmatrix} CB & CAB & CA^2B & CA^3B & \cdots \\ CAB & CA^2B & CA^3B & CA^4B & \cdots \\ CA^2B & CA^3B & CA^4B & CA^5B & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}, \tag{5.23}$$

are computed and sorted in the algebraic order,

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_N. \tag{5.24}$$

The states associated with larger Hankel singular values, $\sigma_1, \sigma_2, \cdots, \sigma_r$, are kept while those associated with smaller Hankel singular values, $\sigma_{r+1}, \sigma_{r+2}, \cdots, \sigma_N$, are truncated. The truncation cut-off index r is determined according to the criterion,

$$\sum_{i=r+1}^{N} \sigma_i < \varepsilon \sum_{i=1}^{N} \sigma_i, \tag{5.25}$$

where ε is a prescribed tolerance. The reduced-order state-space system will be used as the system model for control synthesis.

5.3.2 Subspace Method

An alternative approach to estimate the state-space model using input-output data is the subspace identification method [70,91]. In this method, the linear models are obtained from row and column spaces of certain matrices, calculated from input-output data.

The subspace identification starts by forming the following data matrices [91],

$$\begin{cases} U_p = [u^p_{ij}], \\ U_f = [u^f_{ij}], \\ Y_p = [y^p_{ij}], \\ Y_f = [y^f_{ij}], \end{cases} \tag{5.26}$$

using input-output data sequences, where $u^p_{ij} = u(i+j-1)$, $u^f_{ij} = u(i+j+N-1)$, $y^p_{ij} = y(i+j-1)$, $y^f_{ij} = y(i+j+N-1)$ for $1 \le i \le T_f - 2N + 1$ and $1 \le j \le N$. Then the QR-factorization of matrix $M = [U_f \mid U_p \mid Y_f]$ is computed:

$$M = QR, \tag{5.27}$$

where Q is unitary and R is upper-triangular [31]. Matrix Q can be divided into three column blocks

$$Q = \begin{bmatrix} Q_1 & Q_2 & Q_3 \end{bmatrix}, \tag{5.28}$$

in which the dimension of each Q_i $(i = 1, 2, 3)$ is $(T_f - 2N + 1) \times N$. The upper-triangular matrix R has the form,

$$R = \begin{bmatrix} R_{13} & R_{23} & R_{33} \\ 0 & R_{22} & R_{23} \\ 0 & 0 & R_{33} \end{bmatrix}, \tag{5.29}$$

where the dimension of each R_{ij} $(i, j = 1, 2, 3)$ is $N \times N$. It can be shown that the least square estimate of the Hankel matrix, H, defined as

$$H = \begin{bmatrix} CA^{N-1}B & CA^{N-2}B & CA^{N-3}B & \cdots & CB \\ CA^{N}B & CA^{N-1}B & CA^{N-2}B & \cdots & CAB \\ \vdots & \vdots & \vdots & \cdots & CA^2B \\ CA^{2N-2}B & CA^{2N-1}B & \cdots & \cdots & CA^{N-1}B \end{bmatrix}, \tag{5.30}$$

can be computed by solving

$$R_{22}H^T = R_{23}. \tag{5.31}$$

The order of the system model, r, can be determined using the singular values of H using an approach similar to the balanced truncation method described in Section 5.3.1 based on the singular value decomposition of H,

$$H = U\Sigma V^*. \tag{5.32}$$

The columns of U and V corresponding to singular values $\Sigma_r = diag\{\sigma_1, \sigma_2, \cdots, \sigma_r\}$, denoted by U_r and V_r, respectively, are used to form the finite-interval observation Gramian $\tilde{O}(r)$ and the controllability Gramian, $\tilde{P}(r)$,

$$\tilde{O}(r) = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{r-1} \end{bmatrix} = U_r \Sigma_r^{1/2}, \tag{5.33}$$

$$\tilde{P}(r) = \begin{bmatrix} B & AB & \cdots & A^{r-1}B \end{bmatrix} = \Sigma_r^{1/2} V_r^*, \tag{5.34}$$

from which the state space realization (A, B, C, D) can be calculated. It can be shown that the least-square estimate of matrix A can be found by solving

$$\tilde{O}(r-1)A = \tilde{O}_p(r), \tag{5.35}$$

where

$$\tilde{O}_p(r) = \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^{r-1} \end{bmatrix}. \tag{5.36}$$

Matrices B and C can be obtained from the first row block and column block of $\tilde{O}(r)$ and $\tilde{P}(r)$, respectively [57].

5.3.3 Comparison of Identification Methods

For purposes of validation and accuracy check, the system identification methods described in Section 5.3.1 and Section 5.3.2 are used to identify a known discrete-time linear system under the influence of noise. The chosen linear system has the following input-output expression:

$$\sum_{t=0}^{n} b_i \xi(t-i) = \sum_{t=0}^{n} a_{i,k} u(t-i), \tag{5.37}$$

where ξ is the output and u is the input and the coefficients b_i and $a_{i,k}$ are

$$b = \begin{bmatrix} 0 & 0 & 2 & 1.7 & 1.8 \\ 0 & 0.4243 & 0.3818 & 0.4879 & 0 \end{bmatrix}, \tag{5.38}$$

$$a = \begin{bmatrix} 1 & 1.7 & 2.52 & 1.53 & 0.81 \end{bmatrix}. \tag{5.39}$$

The four poles of the above linear system are known to be:

$$
\begin{aligned}
&-0.40 + 0.86023252i, \\
&-0.40 - 0.86023252i, \\
&-0.45 + 0.83516465i, \\
&-0.45 - 0.83516465i.
\end{aligned}
$$

In order to simulate the influence of noise in the system, each output channel ξ is superposed by a zero-mean white noise sequence $\delta\xi$ to produce the total output (to be used for system identification):

$$y = \xi + \delta\xi, \tag{5.40}$$

The superposed noise can be viewed as the measurement noise, or the effect of plant noise measured at the output channel. The noise sequence $\delta\xi$ is scaled to satisfy

$$|\delta\xi| = r|\xi|, \tag{5.41}$$

where $|\cdot|$ is the vector 1-norm and r is the level of noise relative to the true system output ξ.

The (noisy) input-output data sequence generated by this system is used to identify the original linear system using the two identification methods described in previous sections. Since the original system is known exactly, the accuracy of the identification methods can be determined based on the differences between the identified systems and the original system.

The computational setup follows. The value of r is varied from 0.02 to 0.5 to simulate different noise levels. The input sequence u is zero-mean white noise. Two input-output data lengths, 5,000 points and 50,000 points, are chosen to represent "short" and "long" data sequences. The noise levels are used to test the sensitivity of identification methods for data contaminated by noise, and the data lengths are used to test the convergence of the identified system parameters. For each case, 100 runs were carried out, using independent zero-mean sequences for u and $\delta\xi$ in each run. For the ARX model, a tenth order system is first identified, and a balanced truncation model reduction method is used to reduce the system to fourth order. For the subspace method, the identified system is also truncated to fourth order after state-space balancing. The final system order is determined by the magnitude of significant Hankel singular values of the balanced systems.

We first focus on the identification of the pole $-0.45 + 0.83516465i$ using 5,000 input-output data points. Figure 5.1(a) and Figure 5.2(a) show the identified poles from each run, using the ARX method and the subspace method, respectively, with noise level $r = 0.02$, and the real pole, indicated by the intersection of the horizontal and vertical line segments at the center of the plot. The abscissa and ordinate are the real and imaginary parts, respectively, of the poles. It is seen that both methods produce fairly accurate results at this noise level.

When the noise level is increased, the ARX method produces more scattering and bias, as shown in Figure 5.1(b)–(d) for noise levels $r = 0.1$, $r = 0.2$ and $r = 0.5$. The identified poles using subspace method also show scattering but have little bias even when the noise level is increased, as shown in Figure 5.2(b)–(d).

In another set of runs, the number of input-output data points were increased to 50,000 in order to examine the convergence of scattering with more data samples. The scattering of the identified pole using ARX method is reduced, but the bias still exists, especially when the noise level is high, as shown in Figure 5.3. The identified poles using the subspace method show less scatting and less bias, as seen in Figure 5.4. The identified of the other three poles show similar patterns.

To quantify the overall identification error, the relative identification error, defined to be the difference between the norm of the identified system and the norm of the true system normalized

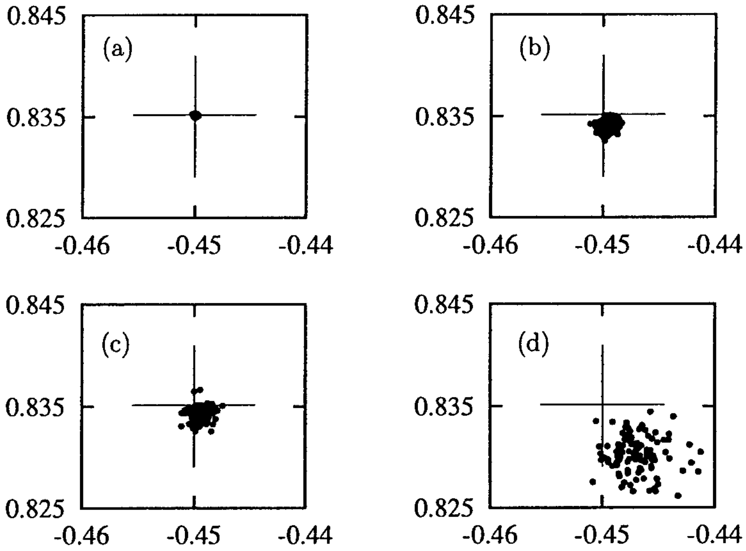

Figure 5.1: The identified pole using least-square estimate of the ARX model with 5,000 data points: (a) $r = 0.02$ (b) $r = 0.1$ (c) $r = 0.2$ (d) $r = 0.5$.

by the norm of the true system,

$$\frac{||S_{id}||_2 - ||S_{true}||_2}{||S_{true}||_2}, \tag{5.42}$$

against various noise levels is plotted in Figure 5.5. Consistent with previous results, the relative identification error increases more rapidly for the ARX method than that of the subspace method. Therefore, it is concluded that the subspace identification is more accurate than the ARX method for a system influenced by noise. Therefore, the subspace identification method will be used to identify system parameters in all subsequent computations.

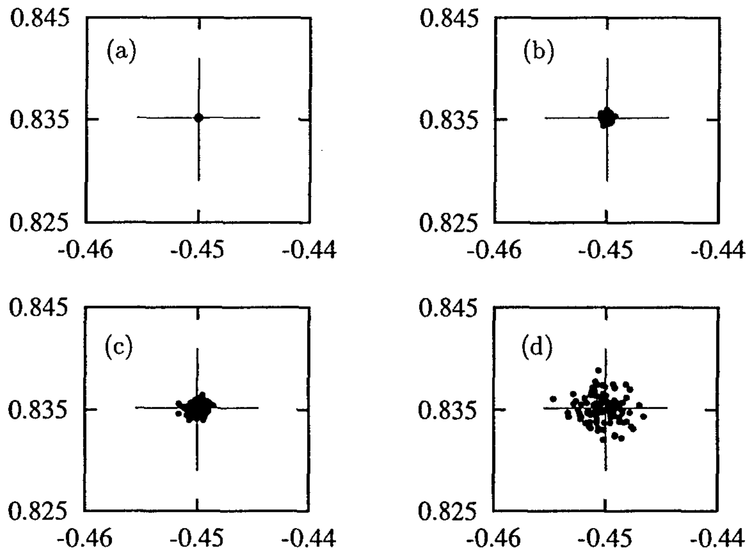

Figure 5.2: The identified pole using subspace method with 5,000 data points: (a) $r = 0.02$ (b) $r = 0.1$ (c) $r = 0.2$ (d) $r = 0.5$.

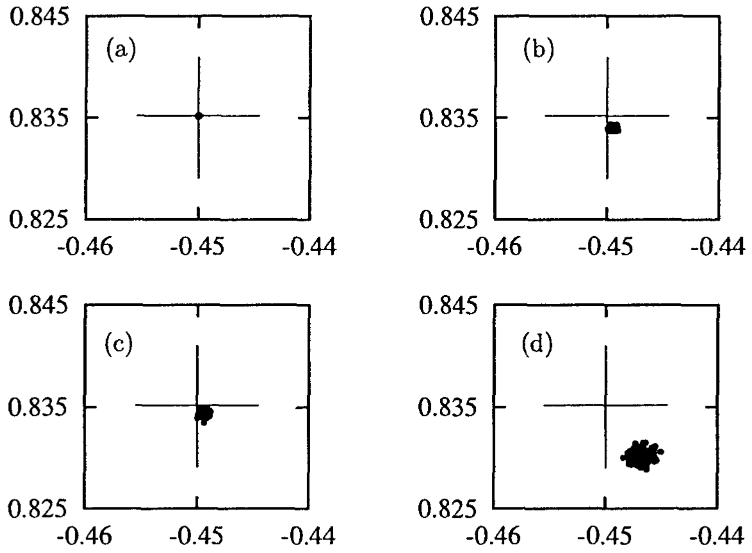

Figure 5.3: The identified pole using least-square estimate of the ARX model with 50,000 data points: (a) $r = 0.02$ (b) $r = 0.1$ (c) $r = 0.2$ (d) $r = 0.5$.

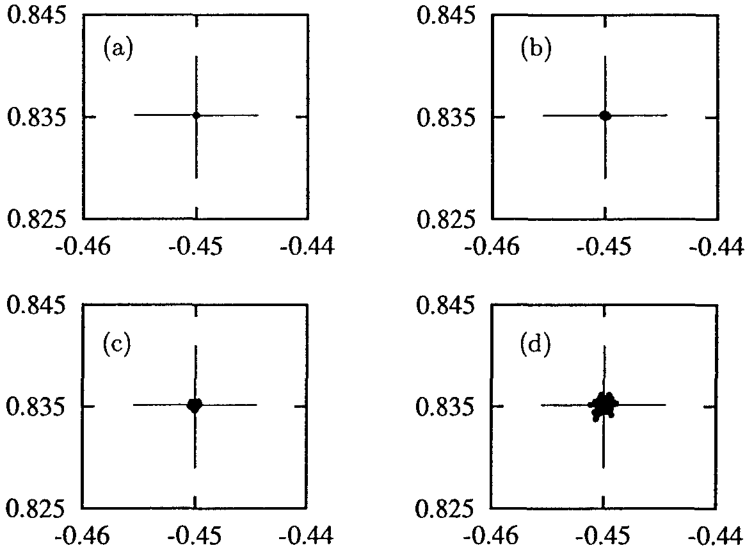

Figure 5.4: The identified pole using subspace method with 50,000 data points: (a) $r = 0.02$ (b) $r = 0.1$ (c) $r = 0.2$ (d) $r = 0.5$.

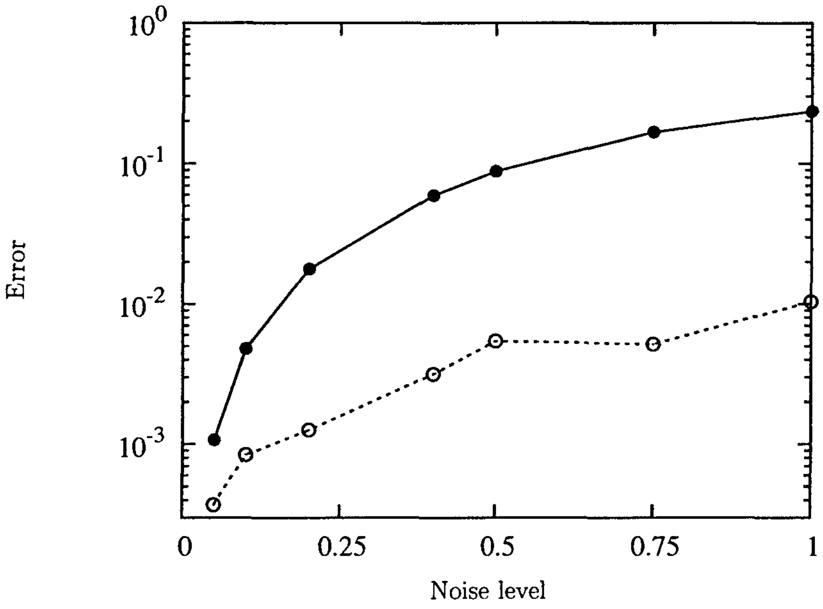

Figure 5.5: Identification error with various noise levels: —— ARX method ---- subspace method.

5.4 Optimal Control

Once a linear system model is obtained from the system identification procedures described in section 5.3, it can be used for control synthesis. Here we apply linear optimal control theory to obtain the optimal feedback law. Note that the feedback law is only optimal for the time-invariant linear system model with respect to a certain (pre-defined) cost function. It is unlikely that the feedback law constructed this way is optimal for the real (nonlinear) flow system, particularly for a simplified linear system model identified using pre-determined input-measurement locations. Nevertheless, existing linear optimal control theory in the literature provides a convenient framework for control synthesis. More accurate system model and parameter estimation procedures may be used within the same framework. In the following sections, we summarize pertinent results from linear optimal control theory that will be used for separated flow control. More detailed treatment of linear optimal control theory can be found in the literature [24,32].

5.4.1 LQR Synthesis

Consider the discrete-time finite-dimensional time-invariant linear dynamical system (5.2). The standard discrete-time LQR (Linear Quadratic Regression) problem is to find the optimal input (or control) sequence $u(t)$ such that the cost function J on the infinite time interval,

$$J = \sum_{t=0}^{\infty} x(t)^* R x(t) + u(t)^* Q u(t), \tag{5.43}$$

is minimized for certain weighting matrices R and Q. When the system (A, B) is stabilizable and (A, Q) is detectable, the solution of the LQR problem is the optimal control sequences

$$u(t) = -Kx(t), \tag{5.44}$$

where the control gain matrix K is

$$K = (R + B^T P B)^{-1} B^T P A, \tag{5.45}$$

where P is the nonnegative symmetric real matrix satisfying the algebraic Riccati equation,

$$P = Q + A^T P A - A^T P B (R + B^T P B)^{-1} B^T P A. \tag{5.46}$$

According to equation (5.44), computing the control sequence $u(t)$ requires the state vector $x(t)$, but this information is not available from the plant (*i.e.*, the separated flow) as discussed in section 5.3. The state vector $x(t)$ is merely a working variable used for deriving a linear model of the plant. It has no obvious physical meaning, nor is available for feedback. The only available information from the plant is the measurement sequence $y(t)$. The Kalman filter [4] provides a way to compute an optimal state estimate, which can be used to compute the control sequence $u(t)$, based on the measurement $y(t)$. Computing the feedback control sequence $u(t)$ using state estimate is known as the LQG (Linear Quadratic Gaussian) problem, described in the next section.

5.4.2 LQG Synthesis

In the LQG (Linear Quadratic Gaussian) problem, one looks for the optimal feedback control gain as well as an optimal filter for state estimate. According to the separation principle [32], these two problems can be solved independently. Since the optimal control gain is computed in the same way as the LQR problem discussed in section 5.4.1, we focus on the optimal filter, known as the Kalman filter, in this section.

When a linear system and its measurement are influenced by noise, its state-space representation can be written as,

$$\begin{cases} x(t+1) = Ax(t) + Bu(t) + B^w w(t), \\ \quad y(t) = Cx(t) + Du(t) + v(t), \end{cases} \tag{5.47}$$

where $w(t)$ is the plant noise and $v(t)$ is the sensor noise. The matrix B^w represents how the plant noise $w(t)$ enters the system. When $w(t)$ and $v(t)$ are white and uncorrelated and defined as

$$\begin{aligned} R^{wv} &= \lim_{t\to\infty} \frac{1}{t} \sum_{i=1}^{t} w(t)v^*(t), \\ R^{vv} &= \lim_{t\to\infty} \frac{1}{t} \sum_{i=1}^{t} v(t)v^*(t), \\ R^{ww} &= \lim_{t\to\infty} \frac{1}{t} \sum_{i=1}^{t} w(t)w^*(t), \end{aligned} \tag{5.48}$$

the optimal state estimate $\hat{x}(t+1)$ of $x(t+1)$ using measurement $y(t)$ up to time t is the discrete-time Kalman predictor,

$$\hat{x}(t+1) = A\hat{x}(t) + F[y(t) - \hat{y}(t)] + Bu(t), \tag{5.49}$$

$$\hat{y}(t) = C\hat{x}(t) + Du(t), \tag{5.50}$$

where F is the Kalman filter gain matrix,

$$F = XC^T R_2^{-1}, \tag{5.51}$$

and X is the solution to the algebraic Riccati equation,

$$X = B^w W B^{wT} + AXA^T - XC^T(R_2 + CXC^T)^{-1}CXA^T, \tag{5.52}$$

Matrices R_2 and W in equations (5.51) and (5.52) are those defined in equation (5.54). It can be shown that the estimation error, defined as $\Delta x(t) = \hat{x}(t) - x(t)$, can be expressed in terms of the impulse response of the transfer function from plant noise to estimation error, g_1, and impulse response of the transfer function from sensor noise to estimation error, g_2, *i.e.*,

$$\Delta x = g_1^* * w + g_2^* * v. \tag{5.53}$$

The state estimate $\hat{x}$ is optimal in the sense that the cost function J_n defined in terms of the weighted sum of the squares of the ℓ_2 norm of the impulse responses g_1 and g_2,

$$J_n = \sum_{t=0}^{\infty} g_1^*(t) W g_1(t) + g_2(t)^* R_2 g_2(t), \tag{5.54}$$

is minimized.

Central to the computation of the solution of the LQG problem is solving the two algebraic Riccati equations (5.46) and (5.52), which can be accomplished using a generalized eigenvalue approach derived by Van Dooren [22]. The LQR control gain K and Kalman filter gain F are then computed by solving the following systems of linear equations,

$$(R + B^T PB)K = B^T PA, \tag{5.55}$$

and

$$FR_2 = XC^T. \tag{5.56}$$

While the solution to the LQG problem can be obtained as described above, given weighting matrices Q, R (in equation (5.43)), W and R_2 (in equation (5.54)), iterations are needed to yield a useful feedback control gain K. For example, the control actuation has an upper bound for its amplitude (*e.g.,* due to mechanical power limitation), which effectively sets an upper limit on the norm of the control gain K. The LQR problem solved above does not consider this constraint, but the norm of the control gain K can be adjusted by choosing the weighting matrix Q in equation (5.43) to meet some practical device specification. In fact, the choice of weighting matrices Q and R in the cost function (5.43) has broader affects to influence the characteristics of the frequency response of the closed-loop system, and is highly problem-dependent. Similarly, to calculate the Kalman filter gain, it is assumed that the noise variances (5.48) are known. However, the noise variances of the plant are generally not known in advance, as they represent the total effects due to external disturbances and the effects of nonlinear dynamics in the flow, in the view of a linear system model. Therefore, the choice of the weighting matrices in equation (5.54) also involves trial-and-error and is problem-dependent.

Therefore, while the cost function (5.43) defines what to be minimized (so that the control is optimal in that sense), the weighting matrices are often adjusted in control synthesis so that the feedback controller satisfies certain properties and design specifications. From an engineering perspective, what is important appear to be the overall performance of the controller, but not a particular form of cost functions. The implications of this will be discussed further in Chapter 6.

It is known that the Kalman filter (5.49) generates the minimum variance state estimate when the noise is Gaussian. If the noise is not Gaussian, the Kalman filter generates the linear minimum variance state estimate [4]. In either case, strictly speaking, the Kalman filter is only applicable to linear systems. When the plant is nonlinear, one may consider the extended Kalman filter, which uses a linearized approximation about a state estimate, or other more sophisticated nonlinear filtering approaches. Since the Kalman filter provides a simple solution to the state estimate problem given a linear model, it is used in the present study for state estimate for feedback control of separated flows.

Chapter 6

Application to a Separated Boundary Layer

In this chapter, the control-theoretic approach described in Chapter 5 is applied to control a separated boundary layer.

6.1 Preliminary Considerations

When the angle of attack of an airfoil is increased, the time-averaged separation point on the suction side tends to move toward the leading edge as the boundary layer encounters stronger adverse pressure gradients. At a large angle of attack, leading-edge separation becomes a dominant mechanism that essentially determines the lift level, and also the overall aerodynamic performance, of the airfoil. It has been believed and observed that if the size of the leading-edge separation region is reduced, the lift force of the airfoil is increased, at least in a time-averaged sense. Therefore, suppressing the leading-edge separation is of primary concern when feedback control is to be used to improve the airfoil's aerodynamic performance. Given a linear system model, while the LQG-based control-theoretical approach described in Chapter 5 provides a general framework for feedback control design, a number of practical issues need to be addressed before applying the approach to separated airfoil flows.

First, the computational cost is significant if full three-dimensional airfoil flow simulations are used to tune control parameters during control synthesis. For control synthesis, simulation of the plant (*i.e.*, the separated flow) should be computationally tractable in order to facilitate parameter tuning. Such parameter tuning is necessary in order to match certain dynamic properties of the controller, such as robustness and noise rejection [32], with those of the plant for control effectiveness. While it is possible to carry out individual three-dimensional calculations of separated flow past an airfoil using the methods described in Chapter 3, such simulations are computationally too expensive with available computational resources when control synthesis is involved, in which a large number of runs need to be carried out to explore the space of control parameters.

Second, even if the computational cost can be reduced to a manageable level by using the DES approach, described in Chapter 2, which has been shown to have the potential of treating high Reynolds-number turbulent separated flows at low cost (relative to DNS/LES), it is not clear how accurate the flowfield solution is in the near-wall regions, which is affected by control actuation. The current DES approach introduces an interface, which is known to be a source causing inaccurate turbulence statistics (see, *e.g.*, [74] and the results discussed in Chapter 4), dividing the LES region from the (near-wall) RANS region. With control actuation the situation becomes more complicated, as we currently have limited knowledge about the controller constructed based on the approach described in Chapter 5 and its effects on the flow. The combined effects from the modeling error associated with DES and from the feedback control actuation in near-wall regions cause concerns about whether one could distinguish them in complex flows.

Based on the considerations above, for the purpose of posing a separated-flow control problem that is computationally tractable using available computing resources, and to reduce uncertainty in exploring control strategies for separated flows, we chose to use a separated boundary layer as a model flow problem for control design and testing. DNS is used for plant simulation to avoid uncertainties of using DES for the forced flow near the wall. In the following sections, the computational settings of a separated boundary layer are described, and DNS results for a three-dimensional transitional separation bubble are established. These results are used to guide a series of two-dimensional calculations, which are then used to for system identification and feedback control design and testings.

6.2 A Transitional Separation Bubble

A separated boundary layer on a flat plate is created by imposing adverse pressure gradient (APG) to an incoming Blasius boundary layer. When large enough APG is imposed, the boundary layer close to the bottom wall separates and reattaches, forming a separation bubble, a scenario similar to the leading edge separation of an airfoil. In this section, the computational settings of a transitional separation bubble are described and results discussed.

6.2.1 Computational Setup

Adverse pressure gradient is created by applying suction on the top boundary of the flow domain, similar to the approach used by, among others, Alam and Sandham [3] and Spalart and Strelets [86]. This flow can be specified by the suction velocity profile and two Reynolds numbers, $Re_X = XU_\infty/\nu$ and $Re_Y = YU_\infty/\nu$, where U_∞ is the incoming free stream velocity, X the streamwise coordinate of suction measured from the virtual origin of the Blasius boundary layer, Y the wall-normal coordinate where suction is applied, and ν the kinematic viscosity.

The objective of the present study is not to investigate the complete parameter space of this flow, but to investigate whether it is viable to use the system identification approach in designing an effective closed-loop linear controller for separated flows. For this purpose, the Reynolds numbers $Re_X = 10^5$ and $Re_X = 3Re_Y$ are used. On the top boundary of the computational domain, the following velocity boundary conditions are prescribed:

$$\frac{\partial u}{\partial y} = \frac{\partial v}{\partial x}, \tag{6.1}$$

$$v(x) = v_m \exp\left[\frac{-\alpha(x-X)^2}{Y^2}\right], \tag{6.2}$$

$$w = 0, \tag{6.3}$$

where $v_m = 0.71U_\infty$, $\alpha = 17.3$ and $X = 3Y$.

At the inflow plane, the flow is assumed to be a two-dimensional, steady, zero-pressure gradient laminar boundary layer, described by the boundary layer equations

$$\begin{cases} u\dfrac{\partial u}{\partial x}+v\dfrac{\partial u}{\partial y}=\nu\dfrac{\partial^2 u}{\partial y^2}, \\ \dfrac{\partial u}{\partial x}+\dfrac{\partial v}{\partial y}=0, \end{cases} \tag{6.4}$$

with the boundary conditions

$$\begin{cases} u=v=0 \quad \text{at } y=0, \\ u=U_\infty \quad \text{at } y\to\infty. \end{cases} \tag{6.5}$$

By introducing the similarity transformation variable

$$\eta=y\sqrt{\frac{U_\infty}{2\nu x}},$$

where x is the streamwise distance from the virtual origin of the Blasius boundary layer, the boundary layer equations (6.4) can be transformed to the Blasius equation,

$$f'''+ff''=0, \tag{6.6}$$

where $f=f(\eta)$ and the primes denote derivatives with respect to η. In equation (6.6), f is related to the stream function ψ by

$$f=\frac{\psi}{\sqrt{2\nu U_\infty x}}, \tag{6.7}$$

and is subject to the boundary conditions

$$\begin{cases} f'=f=0 \quad \text{at } \eta=0, \\ f'=1 \quad \text{at } \eta\to\infty. \end{cases} \tag{6.8}$$

Equation (6.6) is solved by a shooting method based on a fourth-order Runge-Kutta method, yielding the Cartesian velocity components u and v, in streamwise and wall-normal directions, respectively, within the Blasius boundary layer,

$$u=f'(\eta)U_\infty, \tag{6.9}$$

$$v=(\eta f'-f)\sqrt{\frac{\nu U_\infty}{2x}}, \tag{6.10}$$

which are prescribed at the inflow plane of the computational domain. The spanwise velocity component is assumed to be zero at the inflow plane. It is known that, according to the solution of the Blasius equation, the wall-normal velocity v in equation (6.10) is finite at $\eta\to\infty$. Generally, this is inconsistent with the prescribed suction velocity in the wall-normal direction given in equation (6.2) which decays to zero exponentially (more precisely, decays like $\sim \exp[-x^2]$). As a compromise, at the inflow plane, the streamwise velocity is taken from the Blasius solution using equation (6.9), while the wall-normal velocity is set to zero. This approximation is justified by the observation that as the boundary layer enters the APG region, the flow is affected strongly by the suction where the Blasius boundary layer ceases to exist within short distance.

At the outflow plane, the convective boundary conditions

$$\begin{cases} \dfrac{\partial u}{\partial t}+U_c\dfrac{\partial u}{\partial x}=0, \\ \dfrac{\partial v}{\partial t}+U_c\dfrac{\partial v}{\partial x}=0, \\ \dfrac{\partial w}{\partial t}+U_c\dfrac{\partial w}{\partial x}=0, \end{cases} \tag{6.11}$$

are prescribed, where U_c is a convective velocity adjusted at each time step to allow vortical structures to exit the computational domain without much distortion. If U_c is too small, numerical oscillations can take place near the outflow plane. If U_c is too large, it would limit the time step size since the time derivative terms in equations (6.11) are treated explicitly. Numerical experiments show that a good choice of U_c at the outflow plane is

$$U_c = r \max |u|, \tag{6.12}$$

where the max operation is applied to all streamwise-velocity nodes on the outflow plane, and the scaling factor r is adjusted so that the global mass conservation is satisfied.

Cartesian mesh is used for calculating the flowfield of the separated boundary layer, although the Navier-Stokes equations are solved in their generalized-coordinate form described in Chapter 3. In order to resolve steep velocity gradients in the wall region, the grid points are compressed in the wall-normal direction near the wall. The use of zero-vorticity condition (6.1) along the top boundary of the computational domain allows relatively coarse grid (in the wall-normal direction) to be used there. Using Cartesian mesh is advantageous because the cross-derivative terms in the discrete Poisson equation (3.34) vanish, resulting in faster convergence of the multigrid iterations as opposed to their existence in non-orthogonal mesh used in airfoil flow calculations. When uniform mesh is used along the streamwise direction, the fast transform method described in Section 3.3.2 is used.

6.2.2 Results

The first step is to establish grid-independent solution, since transitional flows are known to be sensitive to grid resolution. Such calculations also provide guidelines for subsequent ones involving feedback control. Results from three different grids are reported here. A companion LES is also carried out for accuracy check. Important grid spacing parameters in these cases are summarized in Table 6.1.

Case	Re_X	N_x	N_y	N_z	Δx	Δz	Type
A	10^5	769	193	192	1.2×10^{-2}	3.1×10^{-3}	DNS
B	10^5	1537	193	192	5.9×10^{-3}	3.1×10^{-3}	DNS
C	10^5	3073	257	256	2.9×10^{-3}	2.3×10^{-3}	DNS
D	10^5	769	129	128	1.2×10^{-2}	4.7×10^{-3}	LES

Table 6.1: Grid spacings of separated boundary layer simulations. Δx is the streamwise grid size. Δz is the spanwise grid size.

Figure 6.1 shows the time-averaged pressure coefficient distribution at the wall. The pressure distributions collapse for all cases in the laminar region and the turbulent region, showing that the solutions are well resolved in these regions. However, there are discrepancies in the transitional region, as expected. Comparing Case A and Case B, as the grid size is reduced, the transition location moves downstream. The pressure distributions of these two cases have same shapes, but differ essentially by a shift in streamwise direction. This suggests a delay of transition in Case B compared with Case A. The pressure distributions of Case B and Case C almost collapses on each other, indicating grid independence. These tests show that the streamwise grid size appears to be most sensitive to resolve the transitional region, consistent with the observations made by Jacobs and Durbin [42] in their by-pass transition simulations. The pressure distribution from the LES

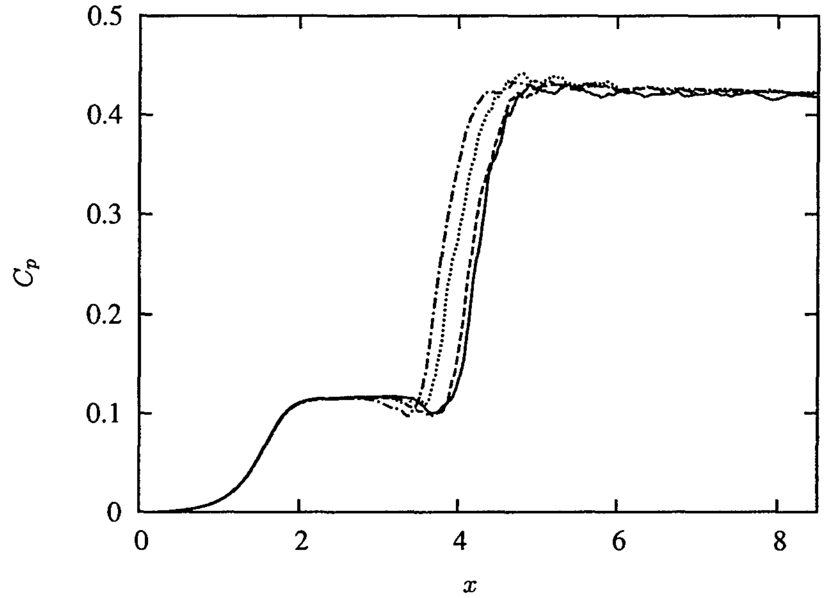

Figure 6.1: Mean pressure coefficient. —·— Case A; ---- Case B; —— Case C; ········ Case D.

calculation (case D) falls between the solution of case A and those of B and C, indicating reasonable accuracy level using LES.

Figure 6.2 shows the mean skin friction coefficient distribution. Similar to the wall pressure distribution, the skin friction distributions are essentially the same in the laminar and turbulent regions for all cases, but differ in the transitional region. In the laminar region, the skin friction initially follows the Blasius solution (6.6),

$$C_{f,Blasius} = \frac{0.664}{\sqrt{Re_x}}, \tag{6.13}$$

but deviates from it after the pressure rises as the flow enters the APG region. The skin friction then falls below zero due to a laminar separation. Beyond the separation point the skin friction slightly rises and then drops to its minimum value under the separation bubble, followed by an increase due to turbulent reattachment.

Since our focus in this chapter is about identification and control design of separated flows, other results obtained from the DNS of the transitional bubble will be reported elsewhere.

6.2.3 A Two-Dimensional Flow

Although the flowfield of the three-dimensional transitional bubble can be computed using the current parallel code within much shorter wall-clock time than its serial counterpart, the overall computational cost (wall-clock time and number of processors multiplied together) remains very high, since in control synthesis it is necessary to explore the space of control parameters involving a large number of runs. Therefore, to reduce computational cost, it was decided to conduct flow control study on a two-dimensional version of the separated boundary layer discussed in Section 6.2.

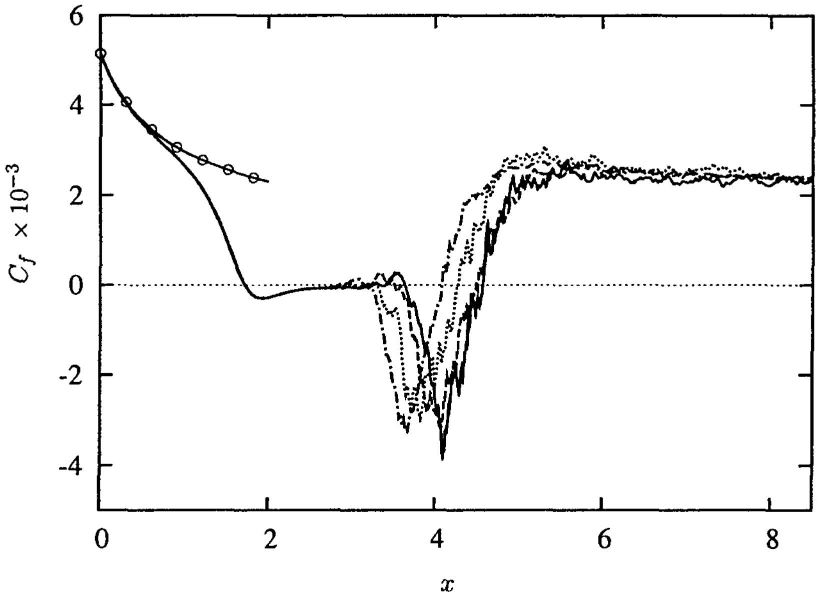

Figure 6.2: Skin friction coefficient. —·— Case A; ---- Case B; —— Case C; ········ Case D; ○ Equation (6.13).

The computational setup is similar to the three-dimensional transitional bubble in Section 6.2, but has only one computational plane in the spanwise direction. The flowfield is two-dimensional, laminar and unsteady. Unlike its three-dimensional counterpart in which the shear layer breaks down and transitions to turbulence, discrete vortices develop beyond the separation region and convect downstream; transition to turbulence is not present due to the two-dimensional constraint.

The Reynolds number used for the subsequent flow control study is $Re_X = 6 \times 10^4$. The computational mesh has 769 points in the streamwise direction, and 192 points in the wall-normal direction. The mesh is clustered around the recirculation region and near the wall to resolve sharp velocity gradients. The domain decomposition described in Chapter 3 is used to carry out the calculations on distributed-memory parallel computers.

The flow is a vortex-shedding separated boundary layer. The similarity of the flat-plate separated boundary layer and the leading edge separation of a NACA0012 airfoil is shown Figure 6.3, in which the instantaneous spanwise vorticity contours are plotted. While this justifies the use of the flat-plate separated boundary layer as a model flow for feedback control design, however, the differences between this two flows should be noted, including, among other things, streamline curvature of the separated shear layer (which for the airfoil is convex, while that of the flat-plate is concave, with respect to the freestream), wall curvature, the effects of trailing edge (which is not present for the flat-plate case).

Figure 6.4 shows the time history of wall pressure at selected measurement locations. The initial flow field at $t = 0$ is a Blasius boundary layer flow across the entire computational domain. Once APG is imposed by suction on the top boundary of the computational domain, vortices are formed in the APG region convecting downstream and the incoming Blasius boundary layer separate from the wall. The flow field for $tU_\infty/Y > 15$ appear to reach a limit cycle, as indicated from the time

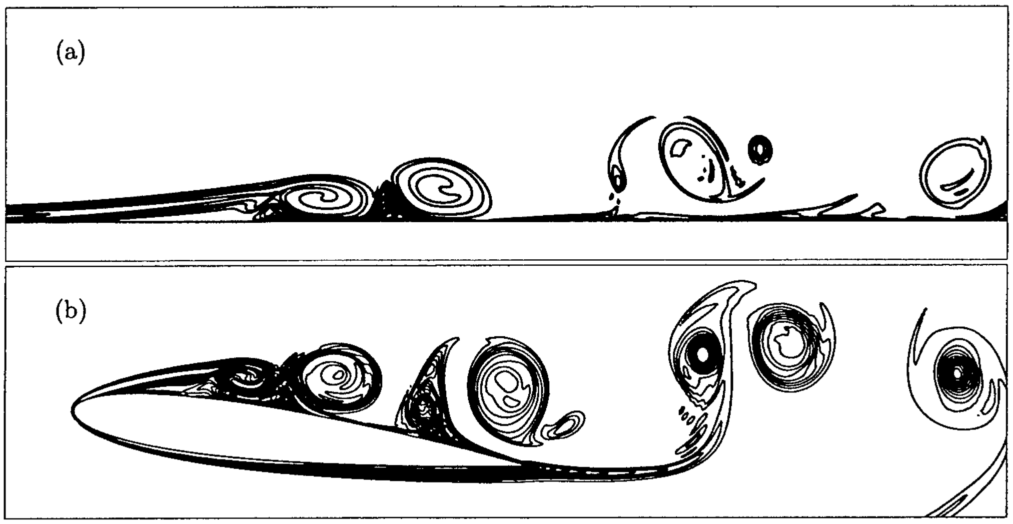

Figure 6.3: Instantaneous spanwise vorticity contours: (a) separated boundary layer, (b) NACA0012 airfoil at 10°.

history of wall pressure. Figure 6.5 shows the time history of spanwise vorticity at same locations.

6.3 Identification of Separation Bubble

A linear model of the separated boundary layer is needed for constructing a linear feedback control of the flow. The system identification approach described in Chapter 5 is used for this purpose. In the following, the computational setup for the system identification of a separation bubble is described, followed by characterization of the identified linear models.

6.3.1 Wall Actuation

Surface blowing and suction are used as control actuation. In all simulations, time-dependent wall-normal velocity components are prescribed to mimic the effects of blowing and suction of real actuators. The internal dynamics of the actuator, which essentially translates the control commands given by the controller to blowing and suction of fluid on the wall, is not considered in this study. Instead, the velocity profile at the blowing and suction location is prescribed according to control commands. The velocity profile at the actuation location is expressed as

$$v_w(x,t) = \phi_1(x)\phi_2(t), \tag{6.14}$$

where $\phi_1(x)$ is the spatial distribution of blowing/suction and ϕ_2 the temporal variation. The function ϕ_1 is expressed as

$$\phi_1(x) = \begin{cases} \dfrac{e^{0.5a} + e^{-0.5a} - e^{-a(x-x_c)/w} - e^{-a(-x+x_c)/w}}{e^{0.5a} + e^{-0.5a} - 2.0}, & \text{if } x_c - \dfrac{w}{2} \le x \le x_c + \dfrac{w}{2}, \\ 0, & \text{otherwise,} \end{cases} \tag{6.15}$$

where w is the width and $x_c = 1.5Y$ is the centerline location of actuation.

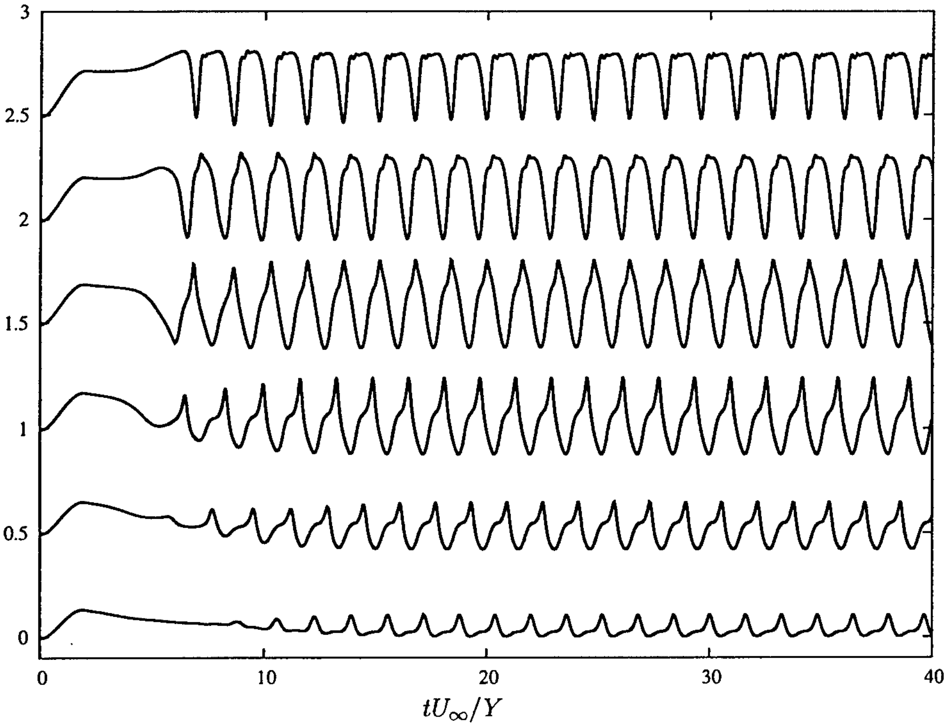

Figure 6.4: Time history of pressure at the wall. From bottom to top: $x/Y = 2.57$, $x/Y = 2.68$, $x/Y = 2.80$, $x/Y = 2.92$, $x/Y = 3.04$ and $x/Y = 3.15$.

The velocity profile in equation (6.15) approaches a parabolic profile when $a \longrightarrow 0$ and approaches a uniform profile when a is large, while maintaining the no-slip condition $\phi_1 = 0$ at $x = x_c - w/2$ and $x = x_c + w/2$. In a real actuator, if the passage between the pumping chamber and the outlet is long enough, the exit velocity would be closer to a parabolic profile. If the passage is short, the velocity profile would be closer to a uniform velocity. However, it should be noted that the prescribed velocity profile is not realistic, especially in the suction phase [53]. Since our attention here is on the identification of separation flows and the design of linear control, more realistic velocity distribution of the actuator is postponed for future study.

The time variation of actuation is determined by ϕ_2. In an open-loop control, ϕ_2 is prescribed (*e.g.*, using predetermined forcing frequencies), while in a closed-loop control, ϕ_2 is determined using information from the flowfield (*e.g.*, wall measurement).

6.3.2 Open-loop Forcing

Open-loop forcing of the separated boundary layer is used for examining a number of features of the forced flow and to create necessary data sequences for system identification calculations.

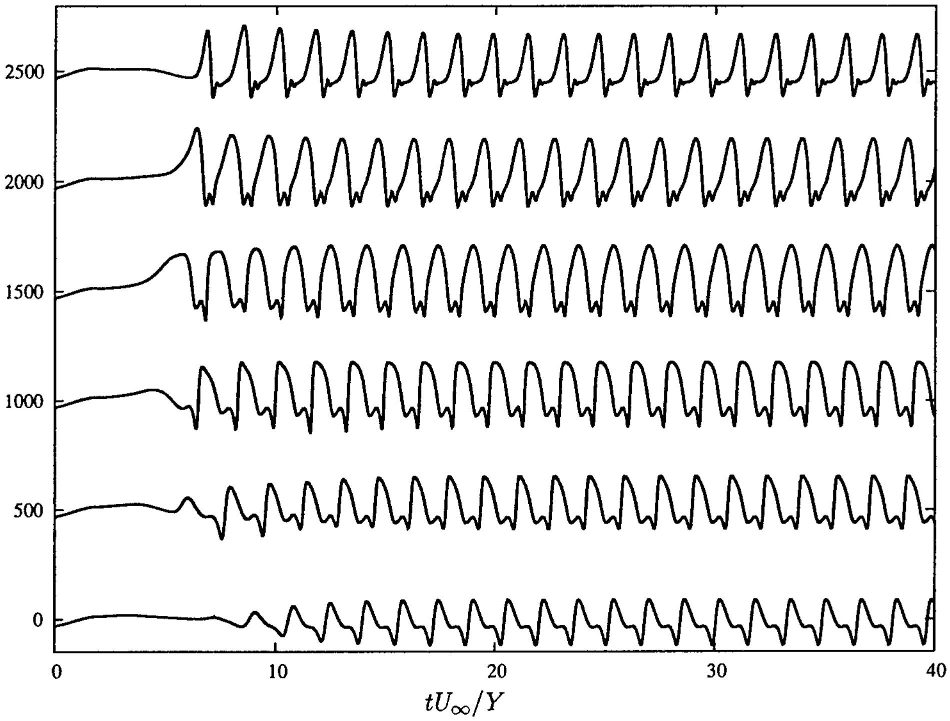

Figure 6.5: Time history of spanwise vorticity at the wall. From bottom to top: $x/Y = 2.57$, $x/Y = 2.68$, $x/Y = 2.80$, $x/Y = 2.92$, $x/Y = 3.04$ and $x/Y = 3.15$.

Single Frequency Forcing

To examine the effectiveness of individual forcing frequencies on separated boundary layer, single-frequency forcing is applied. Specifically, $\phi_2(t)$ in equation (6.14) is chosen to be a sinusoidal function,

$$\phi_2(t) = v_m \sin(2\pi f t), \tag{6.16}$$

where f is the prescribed forcing frequency and $v_m = 0.08U_\infty$ is the maximum forcing magnitude.

A number of nondimensional forcing frequencies f, ranging from 0 to 3, is applied. In each case, starting with an unforced (vortex shedding) flow field as the initial condition, the simulation is first advanced for a time interval of $100tU_\infty/Y$, followed by another time interval of $50tU_\infty/Y$ for calculating flow statistics.

The instantaneous spanwise vorticity fields of the forced and unforced flows are shown in Figure 6.6. In general, the forcing amplifies the instability in the shear layer, causing the cat-eye structure to form before the shear layer breaks down. However, the shear-layer responds differently to different forcing frequencies. It is observed that the shear-layer instability mechanism acts like a band-pass filter; too high or too low forcing frequency has little impact on the time-averaged bubble length. When forcing at the right frequency, the instability in the shear layer is rapidly amplified, inducing large wall-normal velocity fluctuations which enhance the momentum transport in the wall-normal direction. Such wall-normal velocity fluctuations appear to be the key mechanism to reduce the separation bubble size (in the time-averaged sense).

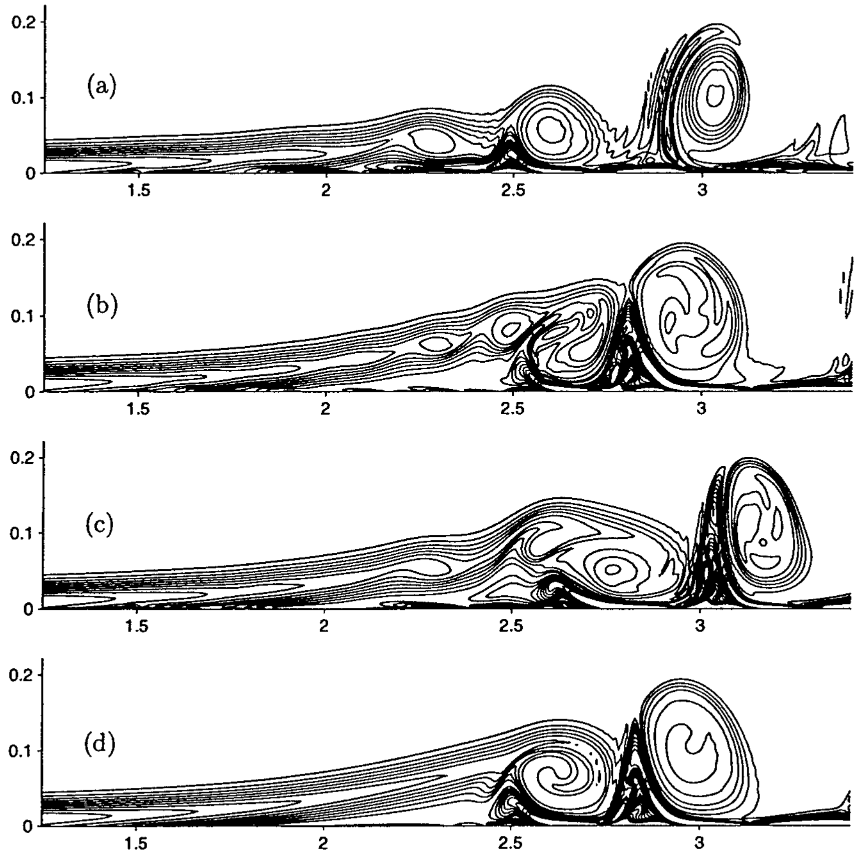

Figure 6.6: Instantaneous spanwise vorticity using open-loop forcing: (a) $f = 1$, (b) $f = 2$, (c) random forcing, (d) no forcing. The scale in wall-normal direction is stretched for clarity.

Station	x/Y
1	2.57
2	2.68
3	2.80
4	2.92
5	3.04
6	3.15

Table 6.2: Streamwise coordinates of measurement stations.

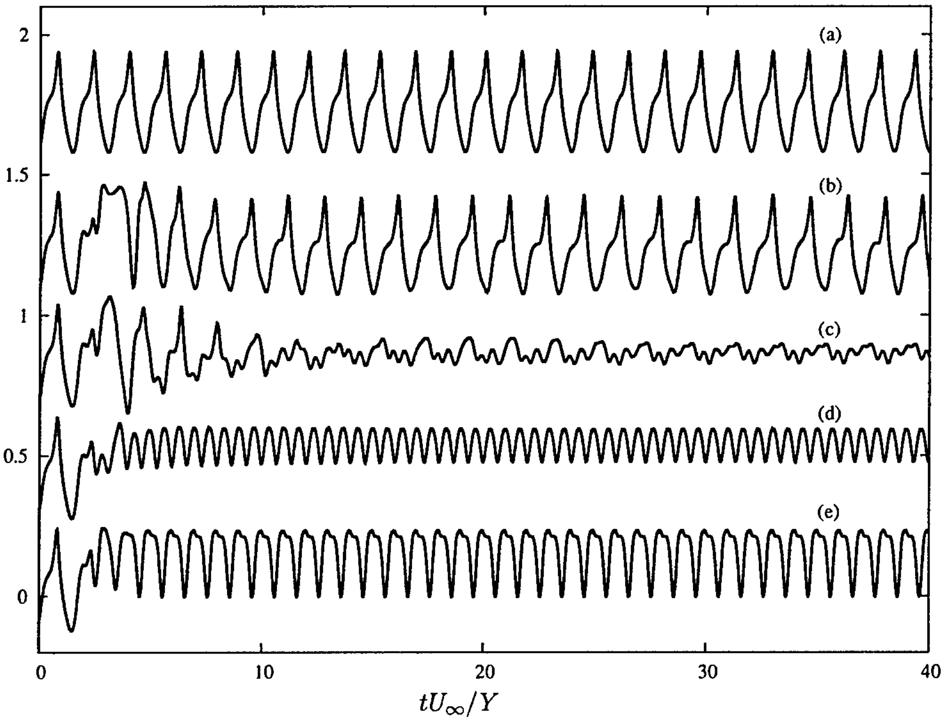

Figure 6.7: Pressure time history of forced flow at measurement station 3: (a) $f = 0$, (b) $f = 2.5$, (c) $f = 2.0$, (d) $f = 1.5$, (e) $f = 1$.

To facilitate subsequent discussion, the streamwise coordinates (relative to the virtual origin of Blasius boundary layer) of measurement stations are listed in Table 6.2. Figure 6.7 shows the pressure time history at measurement station 3 using different forcing frequencies. The initial condition of simulation is an unforced vortex-shedding separated boundary layer. After the initial transient, pressure fluctuations are reduced when forcing is applied. Forcing at a frequency higher than $f = 3$ has little impact on pressure fluctuation measured at this station.

The time-averaged view of the flow fields are presented next. The locations where a time-averaged streamline intersects with the wall are identified as the separation and reattachment points. The time-averaged bubble length is defined to be the distance between the separation and the reattachment points. The dividing streamlines of the time-averaged flowfields for the cases of $f = 0$ (*i.e.*, no forcing) and $f = 1$ are shown in Figure 6.8. It is seen that the time-averaged bubble length and height are reduced when forcing is applied.

It should be mentioned that, for the same forcing frequency, the bubble lengths could be different when different forcing amplitudes are used (not shown here); the relationship appears to be nonlinear. In an extreme case, for example, if suction is very strong, the whole boundary layer could be sucked into the wall when $(U_\infty \delta^*)/(v_w d) < \mathcal{O}(1)$, where U_∞, δ^*, v_w, d are freestream velocity, boundary layer displacement thickness, mean suction velocity, and length of the suction slot, respectively. This extreme case is not pursued in the present study.

Consistent with other studies of forced separated flows, when keeping other parameters the

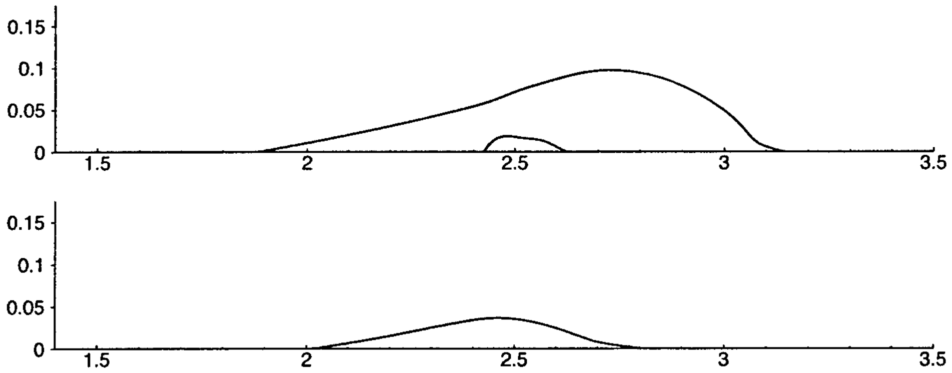

Figure 6.8: Time-averaged zero streamline for the unforced case (top) and that forced at $f = 1$ (bottom.)

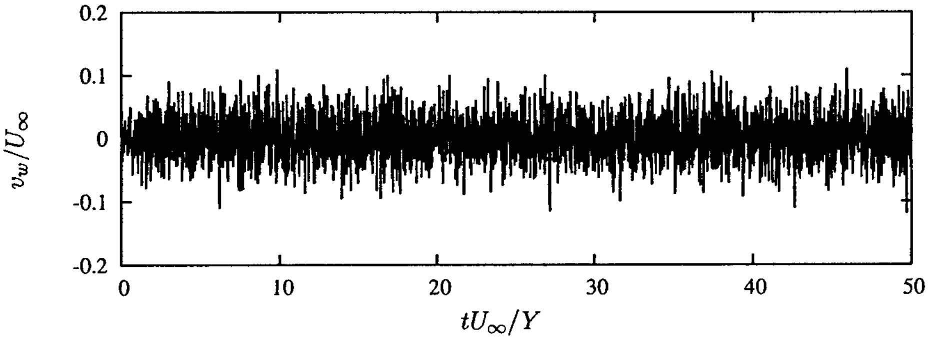

Figure 6.9: White forcing sequence applied to separated boundary layer.

same, there exist an optimal forcing frequency at which the time-averaged separated bubble size reaches a minimum. It would be useful if selecting the optimal frequency can be done automatically by the controller according to the flow state, instead of trial-and-error (usually by sweeping a range of forcing frequencies).

White-noise Forcing

To establish necessary data sequences for system identification, a white forcing sequence is applied at the same actuation location as in single-frequency forcing cases. The time sequences of surface vorticity and pressure at a number of downstream measurement locations are stored for system identification calculations. Starting from an unforced flow field, forcing is applied to the flow over a time interval of 900 tU_∞/Y, corresponding to approximately 300,000 simulation time steps. The duration of forcing is determined by examining the convergence of identified system parameters. The white forcing sequence in the time interval of $[0, 50]$ is shown in Figure 6.9.

The time step size for the separated boundary layer flow simulation is not necessarily the same as the time step in the discrete-time control system. In fact, the time step size for Navier-Stokes simulations is usually smaller than the sampling time of present system identification and discrete-time control. This is because the time step used for solving Navier-Stokes equations has to be small enough for accuracy and stability reasons, discussed in Chapter 3. On the other hand, the sampling time of discrete control is determined by some dominating system dynamics for control purposes. Specifically, the time step size of Navier-Stokes simulations has to be smaller than that of the time scale of small eddies in near-wall recirculation regions in order to obtain correct separated boundary layer evolution, while the sampling time interval in discrete-time control need be only small enough to "resolve" large-scale vortex-shedding that are important for control purposes, which takes place at a larger time scale than that of near-wall small eddies.

Using the input white-noise sequence and recorded data from measurement stations, the linear state-space models of the separated boundary layer flow are computed using the method described in Chapter 5. The results are discussed in the next section.

6.3.3 Linear Models

Two types of linear system models are constructed. The first type is SISO (single-input, single-output) models, based on each input and measurement data pair. The second type is SIMO (single-input, multiple-output) models, based on the same input data and data from multiple output channels. A SISO model represents the input-output relationship corresponding to the actuation and one measurement location. A SIMO model represents the input-output relationship corresponding to the actuation and all measurement stations. Before presenting system identification results, a number of issues regarding the identified models using the present approach are discussed below.

First, the identified model is linear, implying that any nonlinear dynamics of the system cannot be incorporated into the model (more specifically, the system matrix A as discussed in Chapter 5). Since it is known that separated flows exhibit strong nonlinear interactions between the near-wall boundary layer and the inviscid freestream flow, it may not be possible to obtain fully converged system parameter results based on a linear model structure. A central question to this problem is whether the linear model can capture important dynamics of the (nonlinear) system so that an effective control can be constructed.

Second, the identified model only represents the dynamics at the wall, since only wall actuation and wall measurement data are used. This is in contrast to the control-theoretic approach used in other studies (*e.g.*, [10, 55]), in which the linear model, derived from linearized Navier-Stokes equations with respect to a mean flow, contains information inside the flowfield although only wall-measurements are used for feedback.

Third, the identified model is a stable linear system. This implies that the output of the identified model does not grow to infinity no matter how its internal states are perturbed continuously (by either disturbances or actuation). Note that the term "stable" used here is not to be confused with the stable/unstable modes of shear layer stability in fluid mechanics literature. All non-zero states of a stable linear system should decay to zero exponentially without external input/disturbance. However, even if there is no actuation, the present separated boundary layer exhibit persistent vortex shedding, and hence resulting in fluctuating measurement data. This can be explained by viewing the (unforced) separation bubble as a lightly-damped stable linear system subject to continuous external disturbances (due to instability of inviscid-viscous interactions), although the separation of the stable linear part and the disturbances are not carried out explicitly. With this view, it is understood that the system identification of the present separated boundary layer refers to extraction of this stable linear system from measurement data.

Once a linear model is identified, it is useful to check the model's accuracy before using it to design a feedback controller. Ideally model validation can be done by comparing its states or outputs to their counterparts in a corresponding Navier-Stokes simulation upon which the model is based. However, the observations made in previous paragraphs suggest that such direct comparison is not straightforward. For example, since the states in the identified model carry no physical meaning (discussed in Chapter 5), they cannot be compared to any quantity obtained in the corresponding Navier-Stokes simulation. In addition, the flow quantities detected at measurement stations obtained from a forced separated boundary layer are results of the nonlinear interaction of control actuation and the separated boundary layer. The system identification approach can only extract a linear model from measurement data, and treat everything else as disturbances; it cannot distinguish if the disturbance comes from the interaction between actuation and shear layer, or from the nonlinear shear layer dynamics. This makes a direct comparison of system outputs, from the identified model and from Navier-Stokes simulation, respectively, rather difficult.

In order to elicit some meaningful comparison for model validation, the impulse response of the linear model is computed and compared with the output of a separated boundary layer subject to a pulse, detailed in the following section.

Impulse Responses

The impulse responses of the identified linear model corresponding to measurement stations 1 through 6 are shown in Figure 6.10. Each of the linear models based on SISO data sequences uses data from the input sequence and from each of the output channels. The linear model based on SIMO data sequences uses the same input sequence and data from six output channels. When sufficient samples are used, the SISO systems generate essentially the same impulse responses as those from the SIMO system. However, it was generally observed that SISO systems required longer data sequences than the SIMO system in order to achieve converged identification results. Therefore, the identified SIMO linear model is used for all subsequent calculations.

An interesting feature of the impulse responses is that the effect of convection delay is captured by the linear model. After the impulse is applied to the linear model, the model generates a large oscillation, shown in Figure 6.10, at its output channels after a time delay. This appears to be similar to the behavior of a separated boundary layer. When a pulse is introduced at the actuation location in a separated boundary layer, it requires finite time for the pulse to propagate to downstream locations to be detected by sensors at the wall. It is then interesting to see if the time delay observed in the impulse responses of the linear models matches with that of the actual separated boundary layer.

To compare the responses of the linear model and the separated boundary layer, a pulse is introduced at the actuation location in the Navier-Stokes simulations. The pulse is a step function starting at $t = 0$ with duration 0.015 t^*U_∞/Y. The velocity and pressure of a number of trace points are recorded. The coordinates of the trace points are listed in Table 6.3. The time history of the wall-normal velocity components from $t = 0$ to 10 t^*U_∞/Y are shown in Figure 6.11. The influence of the propagation of the pulse to the separated shear layer can be clearly seen.

The approach to compare the impulse response of the identified linear model and the response of the separated boundary layer to a pulse uses the measurement data obtained from the aforementioned pulse simulation. The time history of the difference between the separated boundary layer with and without a pulse at each measurement station is computed and compared with the impulse responses of the identified model. Figure 6.12 shows two such comparisons at measurement stations 2 and 3. The impulse responses of the linear system have been rescaled so that the peak values match with those in the Navier-Stokes simulations. Good agreement is observed at early time, suggesting that the effect of convection delay is captured by the linear model. The deviation

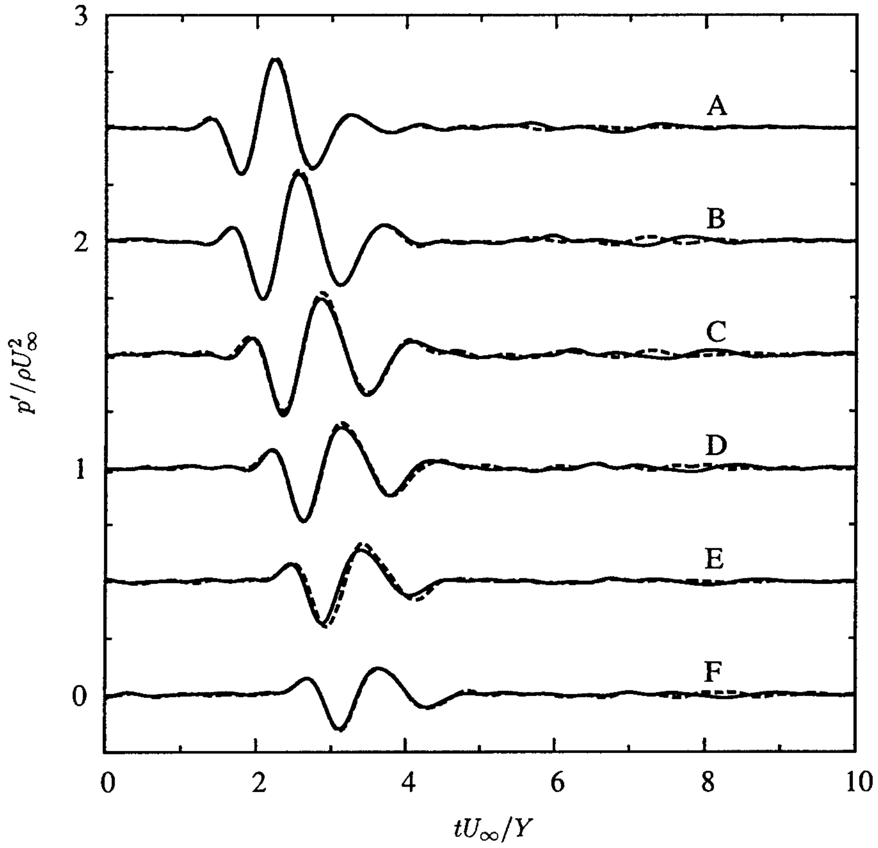

Figure 6.10: Impulse responses of identified models: —— SIMO identification, ---- SISO identification. A: station 1, B: station 2, C: station 3, D: station 4 , E: station 5, F: station 6.

Trace point number	x/Y	y/Y ($\times 10^2$)
1	1.50	1.58
2	1.80	2.13
3	2.00	3.70
4	2.10	4.33
5	2.16	4.99
6	2.21	5.45
7	2.31	6.40
8	2.40	7.40
9	2.51	8.46
10	2.62	8.46
11	2.71	8.46

Table 6.3: Trace point coordinates.

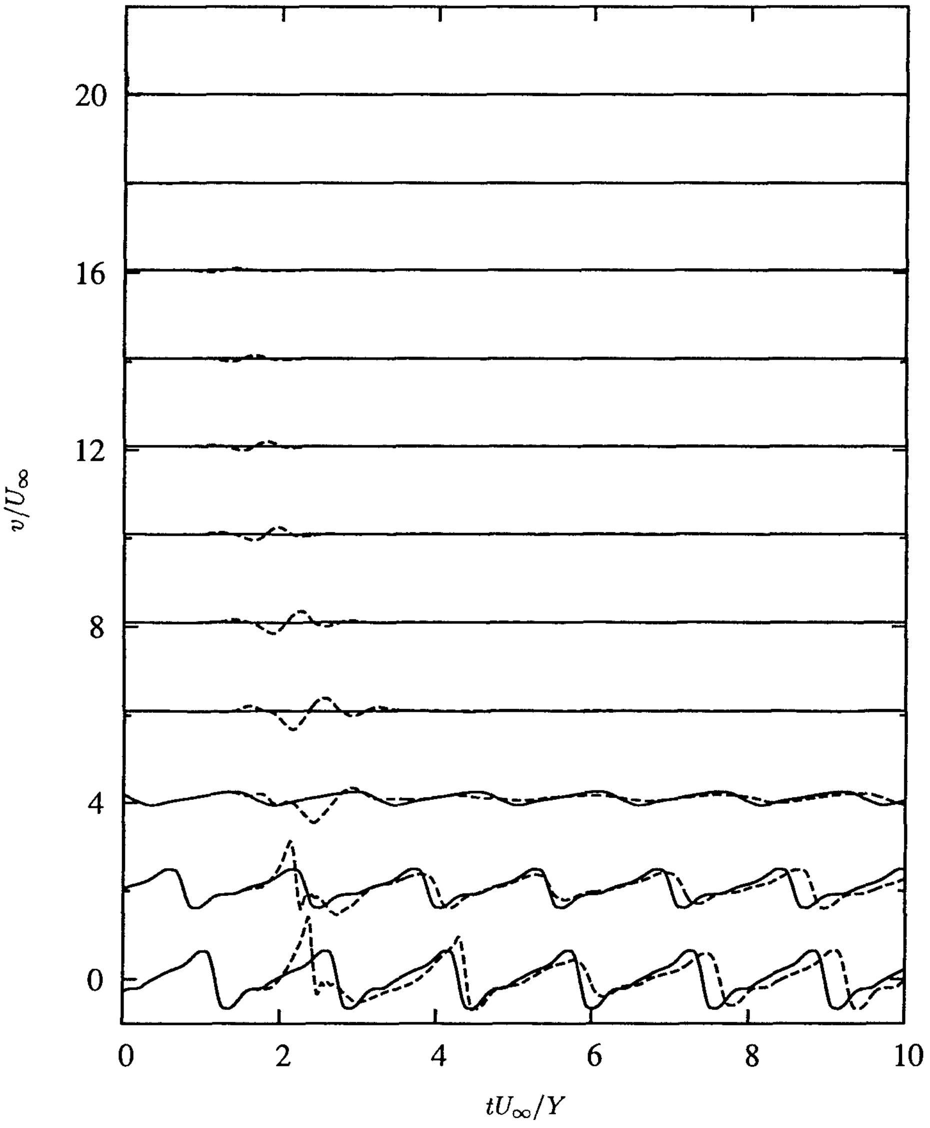

Figure 6.11: Time history of wall-normal velocity on trace points 1 to 11, from top to down in the order of the trace point numbers listed in Table 6.3: —— unforced flow, ---- : flow with step-function pulse. Each time history is offset vertically by 2 units for visualization purposes.

between the two sets of data at later time ($tU_\infty/Y > 4$) are due to the phase difference of the perturbed and unperturbed shear layers due to vortex roll-up.

Frequency Responses

The frequency responses of the linear model corresponding to four measurement locations are shown in Figure 6.13. In Figure 6.13(a), the gain reaches its maximum around $f = 1$, and followed by a drop at $f \approx 2.5$. This suggests that the the identified model would behave like a band-pass filter. Forcing at too high frequency has little effect on the system. This feature is consistent with the observations made in the single-frequency forcing cases discussed in Section 6.3.2. The frequency responses calculated using data from other measurement stations show similar patterns, as seen in Figure 6.13(b)–(d).

Kalman Filter

Once the linear model of the separated boundary layer is available, it is used to construct linear control utilizing linear optimal control theory as described in Chapter 5. In the present feedback control approach, the only available information from the plant (*i.e.*, numerical solution of the Navier-Stokes equations) is the time sequences of output data collected at measurement stations. Thus, a state estimator is needed in order to generate the estimate of internal states, which is then used to generate the control command sequence. For this purpose, a discrete-time Kalman filter based on the identified linear model is constructed using the procedure described in Chapter 5, and is used as the state estimator in closed-loop control.

To test the performance of the state estimator, the same input sequence is supplied to both the Navier-Stokes simulations and the Kalman filter and their outputs are compared, as shown in Figure 6.14. Without reliable estimate of the initial state of the Kalman filter, zero initial state is assumed, resulting in zero initial output (due to zero feed-through term) at $t = 0$. After some initial transient, the output of the Kalman filter quickly approaches the outputs of Navier-Stokes simulation. It is seen that the output of the Kalman filter does not collapse with the output from Navier-Stokes simulation due to disturbances (from a linear system perspective). This test suggests that the present Kalman filter is able to produce reasonably accurate output estimate for the separated boundary layer flows.

6.4 Feedback Control

6.4.1 Cost Function

A cost function must be defined based on which the optimal feedback gain matrix can be computed. The cost function may be directly or indirectly related to the actual control goals. In the present study, however, it is not obvious what the optimal choice of cost function should be to achieve the goal of "suppressing separation." An observation from the open-loop tests performed in Section 6.3.2 is that the pressure fluctuations at certain measurement locations in the forced flow are reduced. Therefore, while other choices are possible, the cost function was chosen to possess the form

$$J = \sum_{t=0}^{\infty} p'(x_m;t)^T p'(x_m;t) + \gamma u^T(t)u(t), \tag{6.17}$$

in LQG synthesis to compute the control gain matrix. The linear controller based on equation (6.17) attempts to minimize the pressure fluctuation at $x = x_m$ as well as the control input. The parameter

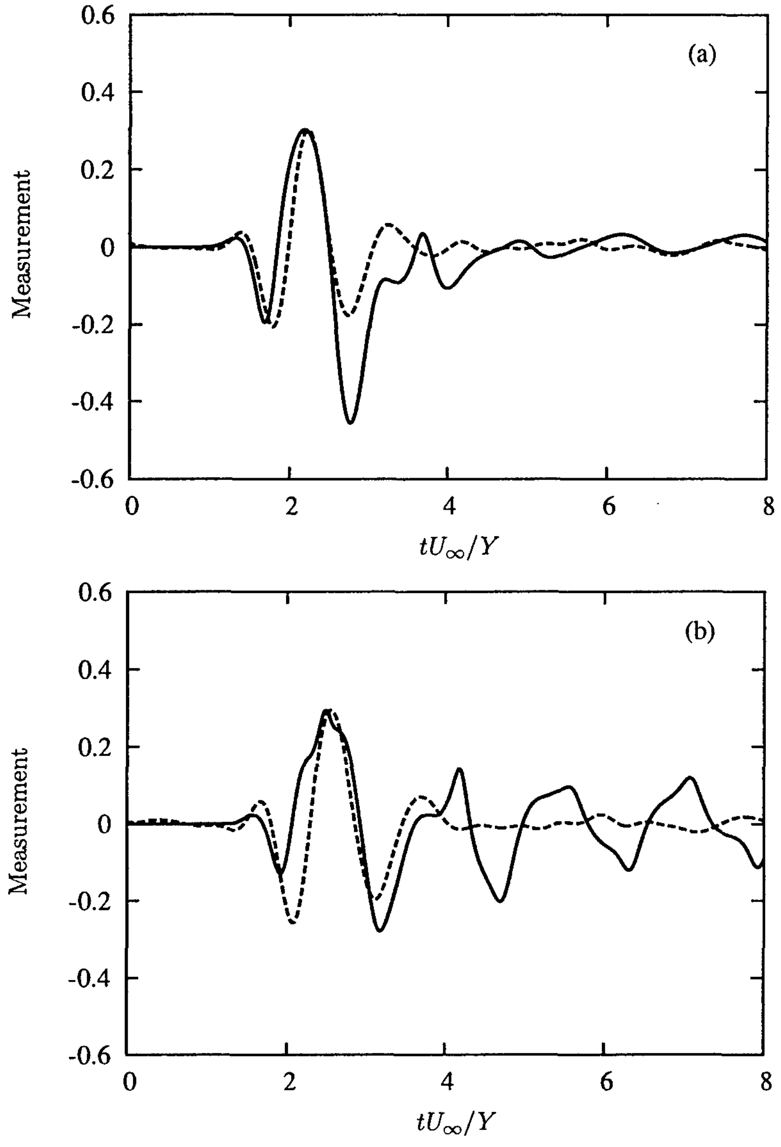

Figure 6.12: Comparison of the impulse response of identified model and the pressure variation of a separated boundary layer with a pulse: —— Pressure difference with and without pulse obtained from Navier-Stokes equation; ---- impulse response of the identified model (scaled): (a) station 1, (b) station 2.

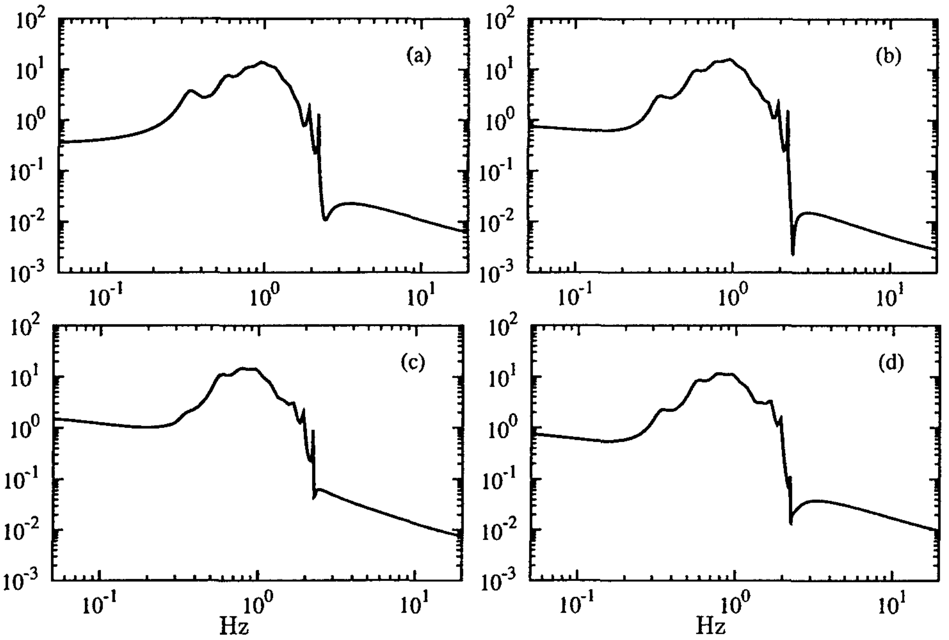

Figure 6.13: Frequency responses of identified models: (a) station 1, (b) station 2, (c) station 3, (d) station 4.

γ in equation (6.17) represents the relative penalty weightings for the cost of control. Increasing the value of γ generally results in smaller control gain, and in turn reduces the blowing and suction intensity of actuation. Reducing the value of γ has an opposite effect.

Once the form of cost function is chosen, the value of γ, together with the noise covariance matrices (see equation (5.54)) are adjusted to achieve desirable control results.

6.4.2 Controlled Flow

A LQG-based feedback control is applied to control separated boundary layer. The initial flowfield before the control is applied is a two-dimensional separated boundary layer. There is no actuation for all $t < 0$. The Kalman filter is initialized to have zero state before control starts. At $t = 0$, the control starts using the pressure fluctuation measurement collected at the measurement stations to generate control command sequence.

Figure 6.15 shows the time history of discrete-time control command. The magnitude of control commands at early time $tU_\infty/Y < 10$ is small, due to the initial response of the state estimator to incoming measurement data. The maximum blowing and suction magnitude gradually reaches approximately $0.1U_\infty$ and appears to saturate after $tU_\infty/Y > 20$. In contrast to single-frequency open-loop forcing discussed in Section 6.3.2, the control command contains multiple frequencies, which are determined based on the identified linear model and the controller rather than predetermined by a trial-and-error process.

Figure 6.16 shows the pressure fluctuation history at measurement stations 2. It is seen that the pressure fluctuations are reduced once feedback control is applied. After some initial transient, pressure fluctuation level reaches a statistically steady state with a reduced amplitude compared

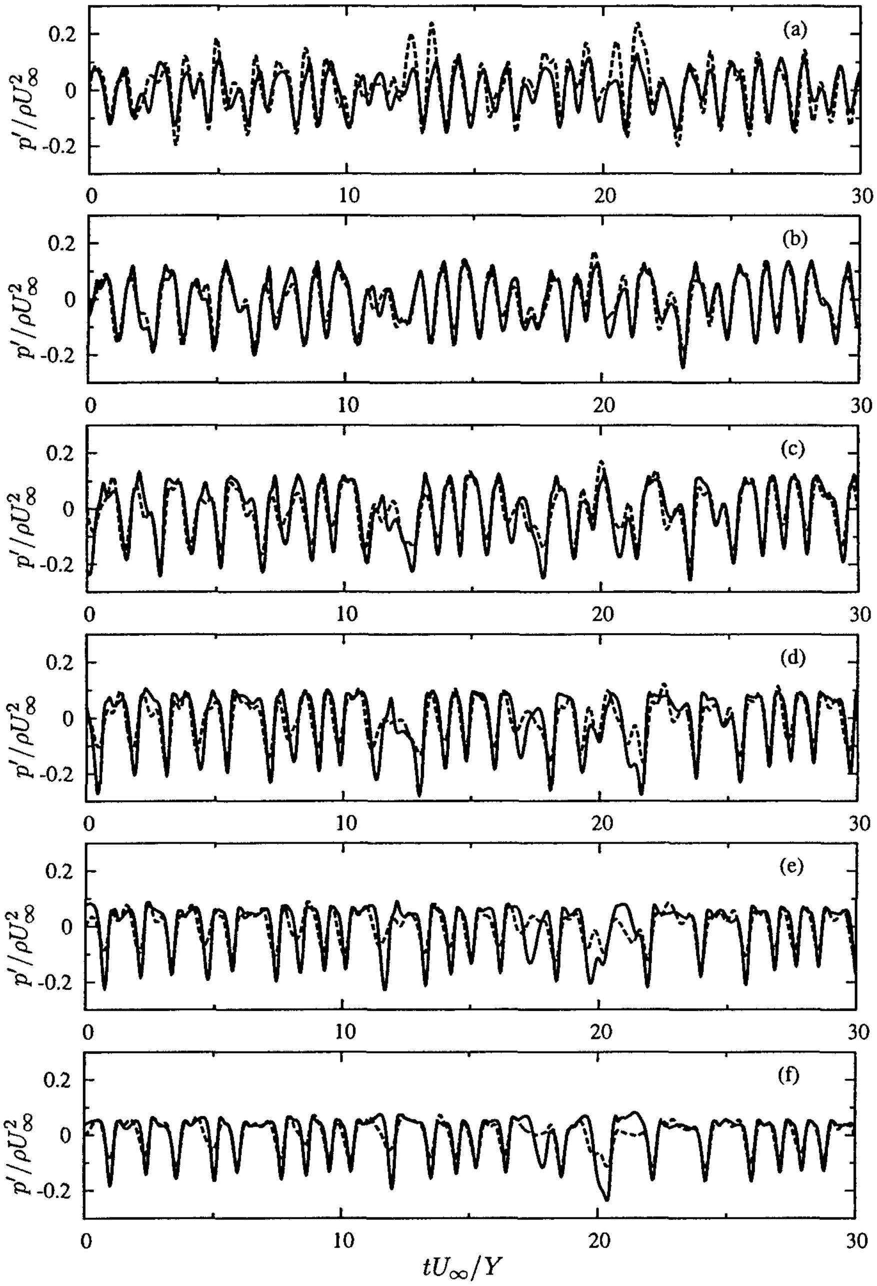

Figure 6.14: Comparison of Kalman filter output and Navier-Stokes simulation: (a) station 1, (b) station 2, (c) station 3, (d) station 4, (e) station 5, (f) station 6, —— Navier-Stokes simulation, ---- Kalman filter.

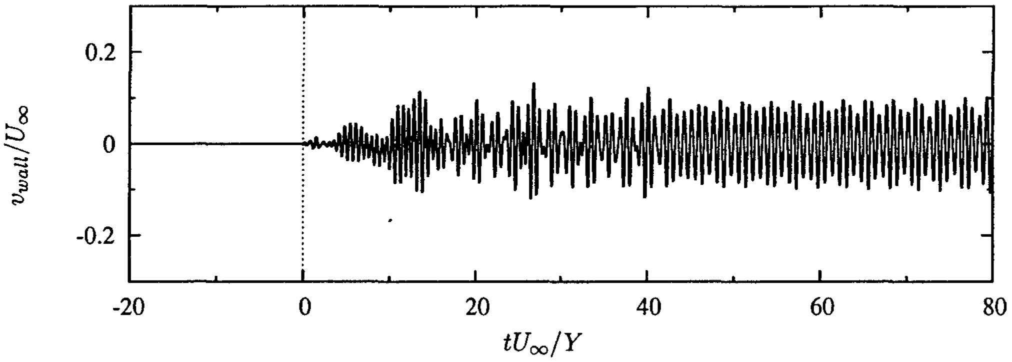

Figure 6.15: Time history of closed-loop control command.

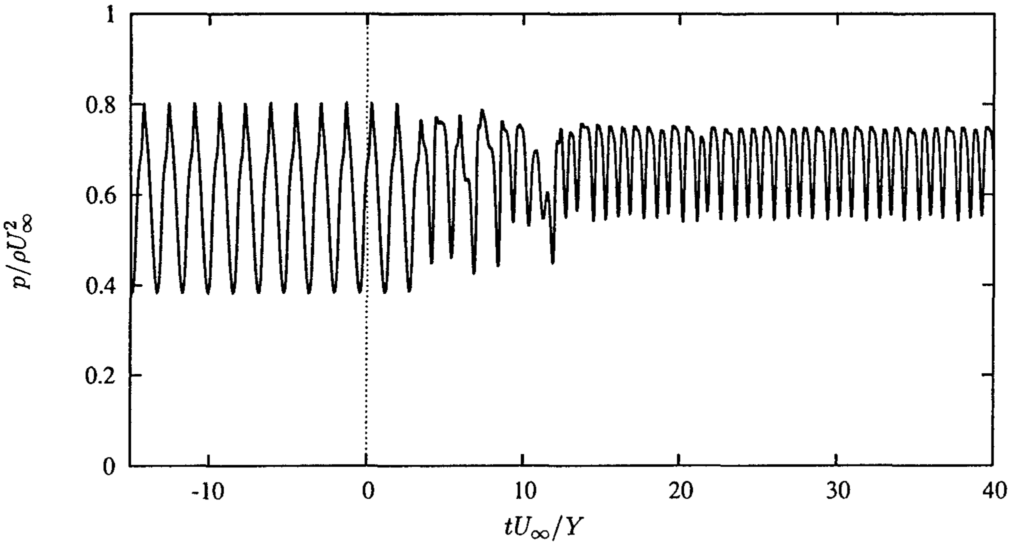

Figure 6.16: Time history of pressure at measurement stations 2.

to its uncontrolled counterpart. It appears unlikely that pressure fluctuations can be made zero by a single upstream actuator (like the present one) due to the existence of viscous-inviscid instability which has its origin away from the wall. Persistent wall pressure fluctuation measurement data also explains why the control commands, shown in Figure 6.15, do not decay to zero.

Figure 6.17 compares the spanwise vorticity fields of unforced and LQG-controlled flows. In controlled flow, the shear-layer is perturbed and shows the cat-eye structures. The perturbed shear layer is closer to the wall, resulting in a smaller separation bubble size.

Figure 6.18 shows the mean streamwise velocity profiles of separated boundary layer with and without feedback control. Since the freestream velocity is varying along the streamwise direction

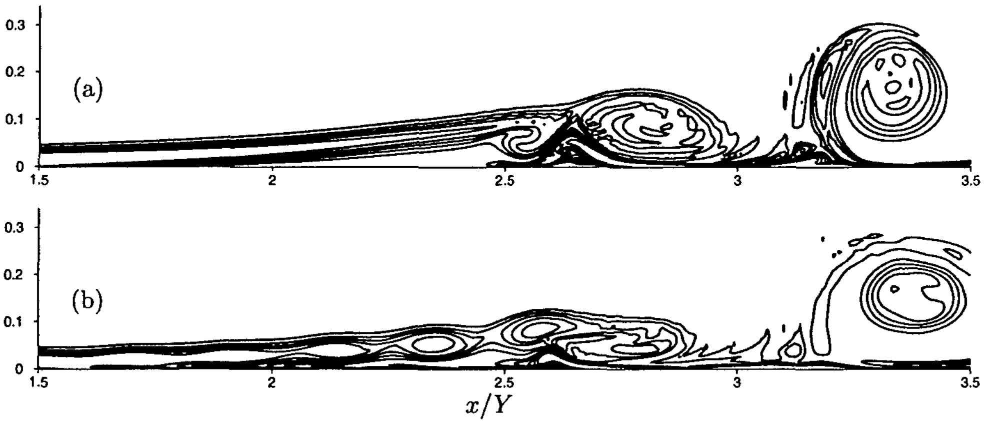

Figure 6.17: Instantaneous spanwise vorticity contours: (a) No control, (b) LQG control.

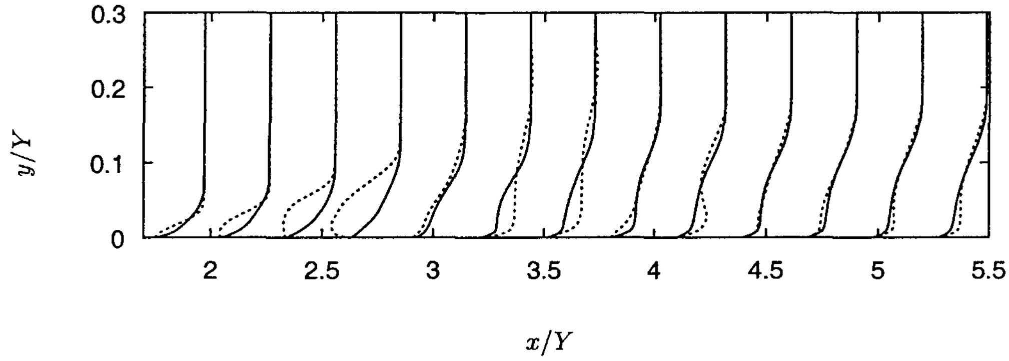

Figure 6.18: Time-averaged velocity profiles of 2D separated boundary layer: —— : uncontrolled; ---- : LQG control.

(due to suction creating APG), the "vorticity velocity", defined in [86]

$$u^* = \int_0^{y_{\max}} \omega_z dy, \tag{6.18}$$

where ω_z is the spanwise vorticity, is used to scale the mean streamwise velocity for each velocity profile in Figure 6.18. It is seen that the reverse flow is largely reduced for the controlled case. The control appears to have little effects on velocity profiles beyond $x/Y > 4.25$.

6.5 Summary

A LQG-based control is constructed for a two-dimensional separated boundary layer. The subspace system identification method is used to construct a linear model based on the input and output data obtained from numerical simulation of the separated boundary layer. The linear model is shown to share a number of features with the separated boundary layer upon which it is based. A Kalman predictor is constructed based on the identified linear model and is used in LQG control synthesis. The cost function in LQG synthesis is based on pressure fluctuations on selected measurement locations, inspired by preliminary tests using open-loop forcing.

The closed-loop control is applied to the two-dimensional separated boundary layer. The control actuation is able to perturb the shear layer at certain frequencies resulting in reduction of time-averaged separation bubble size.

Chapter 7

Conclusions and Recommendations

In the present study, a new parallel computational code which can deal with complex-geometry high-Reynolds number flows is developed and then used to study the control of separated flow. A closed-loop control scheme based on a system identification method and LQG control synthesis has been developed, and is shown to reduce the separation bubble size in a separated boundary layer. A number of more specific discussions that conclude the present study and make recommendations for further directions follow.

7.1 Flow Simulation

An efficient computational code for the simulation of turbulent separated flow utilizing massively parallel computers has been developed, taylored in favor of flow control study. A number of turbulence simulation techniques, including DNS, LES, DES and RANS are integrated within a single numerical framework. Both accuracy and efficiency of the new flow solver have been demonstrated through extensive validations by comparing results for various flows, ranging from turbulent channel flow to separated flow past an airfoil at a large angle of attack, with existing ones found in the literature. Since the flow solver is based on a generalized-coordinate formulation, it is straightforward to deal with other flow geometry to meet long-term research goals.

The parallel algorithm developed in the present study significantly reduces wall-clock run time, compared to its serial counterpart as shown in Chapter 4. The parallelism essentially removes the usual resolution limitation seen in prior turbulent simulations and allows problems of very large size to be solved within relatively short wall-clock time. Specifically, domain decomposition along two coordinate directions are employed, resulting in a quasi one-dimensional problem, in the many-processor limit, to be computed for each processor. This is especially advantageous for turbulence simulations. For example, the domain size along a homogeneous direction can be largely expanded so that sufficient independent samples are available for computing statistics with short simulation time. This is helpful, at least for studying flow control of spatially developing flows and for developing near-wall modeling treatment for LES and DES.

Of particular interest was to test the DES accuracy on wall flows in order to study flow control for higher Reynolds number flows. The calculations carried out in the present study show that, while the DES approach can handle high-Reynolds number flows at lower cost than standard LES approach, it creates an interface between the RANS and LES regions which introduces spurious effects to the flow solution. This issue has to be resolved before the DES approach can be applied to flow control study, in order to isolate control actuation effects from modeling errors. On the other hand, it is observed that the flow solution appears to be less sensitive to the RANS-LES interface for massively separated flows, especially those whose separation point is set by geometry, than for wall-bounded channel flows, since the RANS-LES interface is confined to the vicinity of the wall and appears to have less impact on the separation regions. It was also found that the second-order non-dissipative finite-difference scheme produces more accurate results in the LES region than the upwind-biased schemes, and should be used whenever possible.

7.2 Separation Control

A discrete-time closed-loop linear control scheme is developed for control of separated flows. The linear system model used in the controller is constructed using the input and output data of the separated flow, and is shown to preserve certain important features of the separated flow that it is based on. A closed-loop linear control based on the identified model is shown to be able to reduce the time-averaged separation bubble size of a separated boundary layer.

In order to build a linear model from input and output data, a subspace system identification method is used to construct a linear state-space model for the flow. The subspace method produces more accurate and more robust than an ARX method when noisy data are used. Both SISO (single input, single output) and SIMO (single input, multiple outputs) system models are considered. The main difficulty to using multiple input-output channels is only computational: the model order increases as the number of channels is increased. The high model order is also a consequence of having no Fourier decomposition in any spatial direction, as was used in prior system-theoretic approach for channel flow control studies.

One the linear model of the system is obtained, it is used to construct the state estimator, which is used in LQG control synthesis to produce an optimal feedback control in the infinite time interval corresponding to the choice of cost function. The feedback control is then applied to a separated boundary layer, and is shown to reduce the time-averaged separation bubble size.

While the feedback control scheme based on a linear system model for the nonlinear flow produces promising results, it is clear that there is room for performance improvement, including

- While the closed-loop control determines forcing frequencies based on measurement data and requires no trial-and-error adjustment, the optimal locations of actuation and measurement remains an open question. Future study should address these issues and explore the effects of having multiple actuation locations.

- Other choices of cost function should be explored. For separated flow control, the main control goal is to reduce separation in order to improve the aerodynamic performance of the device in question. However, it was not obvious to define a cost function that can guide the control to achieve this goal. In the future, other types of cost functions should be tested. With the existing parallel code for flow computation, it is expected that such tests may be carried at high speed.

- An adaptive scheme should be explored in order to account for the change in the mean flow after control actuation is applied. The adaptive scheme may improve the accuracy of parameter estimation.

Appendix A

Computation of metric terms

The metric coefficients present in the transformation governing equations are computed and stored at the beginning of a simulation. The accuracy of the metric coefficient has direct impact on the solution's quality. In this study, a fourth-order finite difference scheme and a fourth-order interpolation scheme are used to compute metric coefficients at all staggered locations of flow variables. The details of these computations are documented below.

Differentiation

Metric coefficients are calculated using the same finite difference method for discretizing the convection terms. For example, to evaluate $c_{11} = \partial x/\partial \xi_1$ using the fourth order finite difference method is:

$$\left.\frac{\partial x}{\partial \xi_1}\right|_{x=x_i} \approx \left(\frac{\delta x}{\delta \xi_1}\right)_i = \frac{x_{i-2} - 8x_{i-1} + 8x_{i+1} - x_{i+2}}{12\Delta\xi_1} \tag{A.1}$$

Evaluation of c_{11} using second order finite difference is:

$$\left.\frac{\partial x}{\partial \xi_1}\right|_{x=x_i} \approx \left(\frac{\delta x}{\delta \xi_1}\right)_i = \frac{x_{i+1} - x_{i-1}}{2\Delta\xi_1} \tag{A.2}$$

Interpolation

In this study, a single grid system is generated for each simulation performed. The generated grids correspond to control volume corners used in the computation. To calculate metric coefficients at different variable locations (due to the staggering of velocity nodes), an interpolation scheme is required to evaluate the corresponding grid coordinates from neighboring points available from grid generation. For example, to compute the coordinate x at $(i+\frac{1}{2}, j+\frac{1}{2})$, the following two-dimensional, fourth-order accurate interpolation scheme is used:

$$\begin{aligned} x^*_{i,j} &= \frac{1}{12}x_{i-2,j} + \frac{1}{4}x_{i-1,j} + \frac{1}{4}x_{i+1,j} + \frac{1}{12}x_{i+2,j} \\ x_{i+\frac{1}{2},j+\frac{1}{2}} &= \frac{1}{12}x^*_{i,j-2} + \frac{1}{4}x^*_{i,j-1} + \frac{1}{4}x^*_{i,j+1} + \frac{1}{12}x^*_{i,j+2} \end{aligned} \tag{A.3}$$

The accuracy of the finite difference approximation and interpolation (*e.g.*, Equations (A.1) and (A.3)) can be verified by computing all metric coefficients on a geometry where analytic form of $\mathcal{M}$ is available. When consistent discretization and interpolation schemes are used, the error norm, defined as the absolute difference between the metric coefficients computed numerically and analytically, respectively, should decay and show the corresponding convergence rate as the computational mesh is refined. Figure A.1 demonstrates that fourth-order accuracy, consistent with the interpolation and discretization schemes used here, is achieved for calculating c_{12}. Similar error convergence patterns are obtained for all other metric coefficients. Also shown in Figure A.1 are the errors computed using a second-order accurate formulation in an earlier version of our computer code. It can be clearly seen that the fourth-order formulation used here has not only a higher convergence rate, but also a lower level of error norm than the second-order accurate formulation using the same mesh.

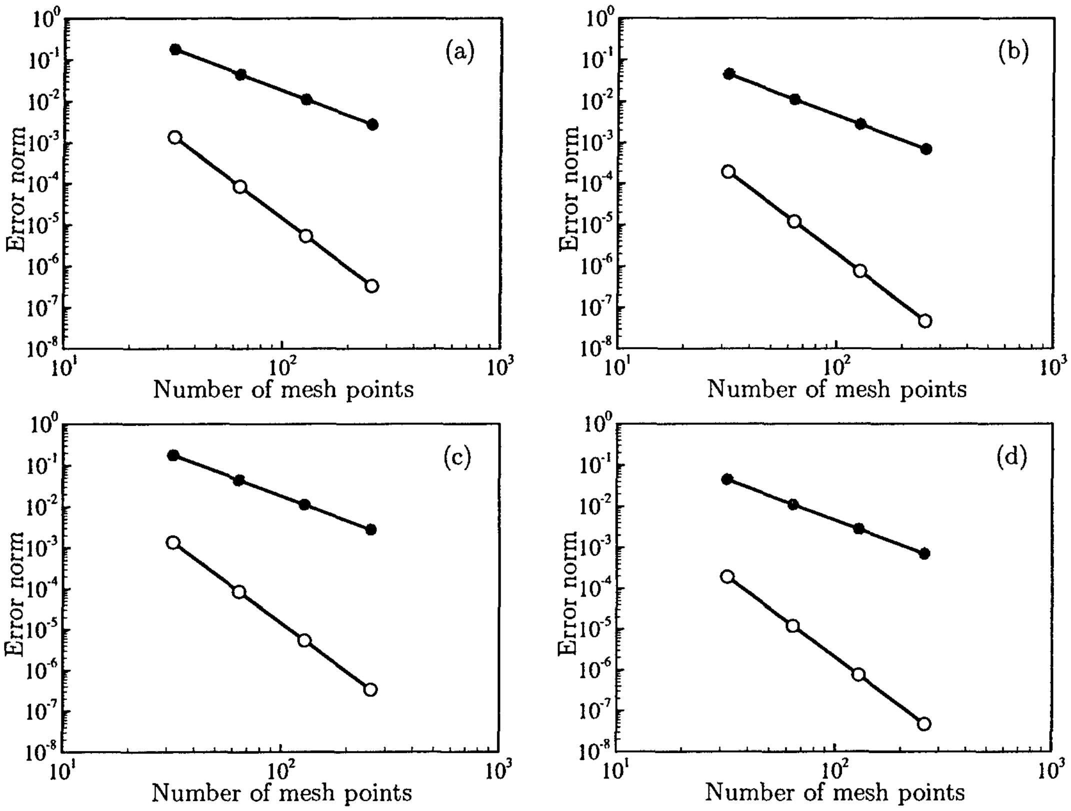

Figure A.1: Error of calculated metric coefficient c_{12}: (a) at location (i,j), (b) at location $(i,j+\frac{1}{2})$, (c) at location $(i+\frac{1}{2},j)$, (d) at location $(i+\frac{1}{2},j+\frac{1}{2})$. ○ fourth-order scheme; ● second-order scheme.

Appendix B

Spurious oscillations

B.1 Effects of grid stretching on centered schemes

Centered schemes are extremely sensitive to grid-to-grid stretching ratio. Spurious grid-to-grid oscillations, or the so-called 2-δ waves, in numerical solution may exist when centered schemes are used. The undesired oscillations sometimes significantly alter the solution, making the numerical solution totally unacceptable. Experience shows that the grid sensitivity of centered scheme is especially severe in directions where flow speed is high. In some cases, the grid-induced 2-δ waves may be minimized by using a uniform or nearly-uniform grid system. The success of previous simulations of turbulent channel flow and flat-plate turbulent boundary layer flow using centered finite difference schemes, even with an under-resolved mesh, may be partly due to the fact that uniform grids are used in the mean (*i.e.*, high-speed) flow directions. For high-Reynolds-number flows in complex geometry, however, economic consideration typically requires the use of highly stretched grids. Centered schemes, despite of their superior non-dissipative property when oscillations do not arise, in many cases fail to produce even a reasonable solution; the onset of spurious oscillations often severely contaminated the solution. Other aspects of using central schemes in high-Reynolds number, complex geometry flow simulations are given by Travin *et al.* [89].

B.2 Spurious Oscillations

When the resolution is marginal, the use of centered schemes, even upwind-biased schemes, sometimes lead to spurious oscillations. This is demonstrated by the results shown in Figure B.1.

B.3 Resolution requirement near the stagnation point

Consider the following model equation,

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \frac{1}{Re}\frac{\partial^2 u}{\partial x^2} \tag{B.1}$$

with boundary conditions

$$\begin{aligned} u = 1 \qquad &\text{at} \qquad x = 0 \\ u = 0 \qquad &\text{at} \qquad x = 1 \end{aligned}$$

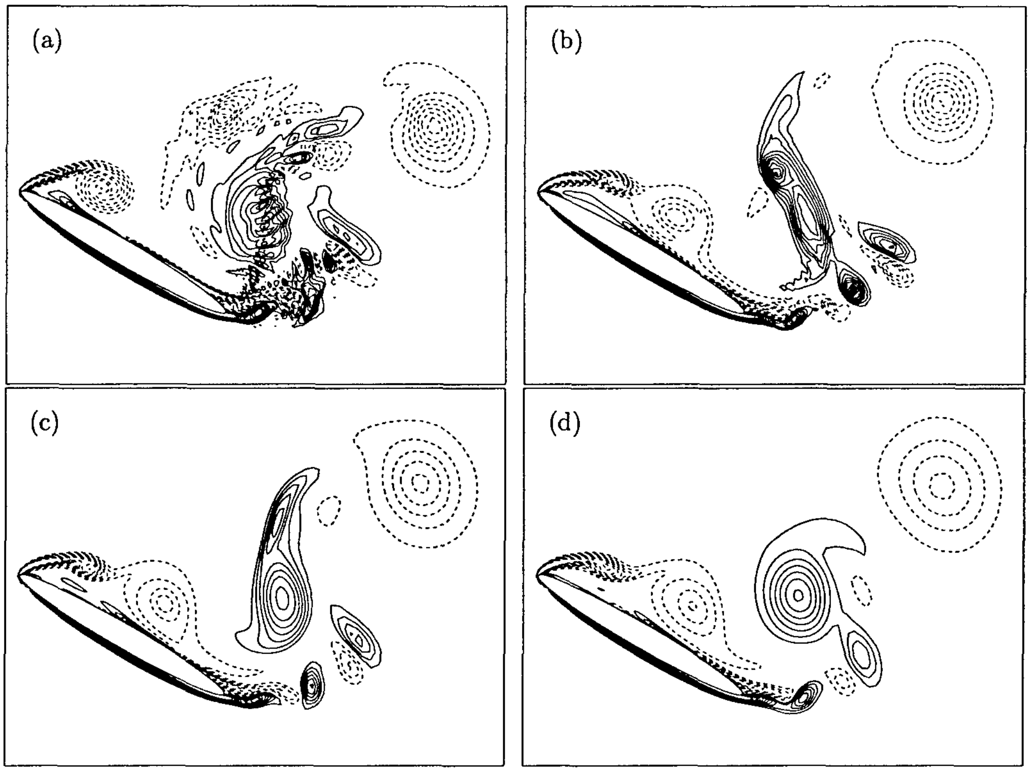

Figure B.1: Effect of finite difference schemes. Spanwise vorticity $-50 \leq \omega_z \leq 50$ with increment of 5. Contours of negative values are dashed: (a) second-order central difference scheme, (b) fourth-order central difference scheme, (c) third-order upwind difference scheme, (d) Kuwahara's upwind scheme.

Note that this model equation does not include pressure gradient term, and its behavior is not expected to be similar to the Navier-Stokes solution for complex flows. Nevertheless, its solution has a boundary layer behavior and is used here for simplicity. In this test, the second-order central difference is used for all spatial derivatives, *e.g.*,

$$\frac{\partial u}{\partial x} \approx \frac{u_{i+1} - u_{i-1}}{x_{i+1} - x_{i-1}} \tag{B.2}$$

A hybrid Crank-Nicolson/low-storage third-order Runge-Kutta method is used for time integration. The grid points are nonuniformly distributed in the interval $[0, 1]$, with more grid points clustered near $x = 1$ to resolve the thin boundary layer, using constant expansion ratio. The solution is advanced in time until steady state is reached. Figure B.3 (a) shows the solution at various Reynolds numbers. Figure B.3 (b) shows the boundary layer thickness δ_{95}, defined as the distance between $x = 1$ and the point where u reaches 0.95. Figure B.3 (c) shows the effects of first grid spacing to the quality of the solution.

It can be seen from Figure B.3(b) that the boundary layer thickness decreases linearly with the Reynolds number. However, this estimate is rather conservative. In the actual airfoil flow near the leading edge, the boundary layer is usually thicker than that estimated here. With the effect of pressure gradient, built up due to the existence of solid wall, velocity components start "slowing down" further away from the wall, leading to a thicker boundary layer.

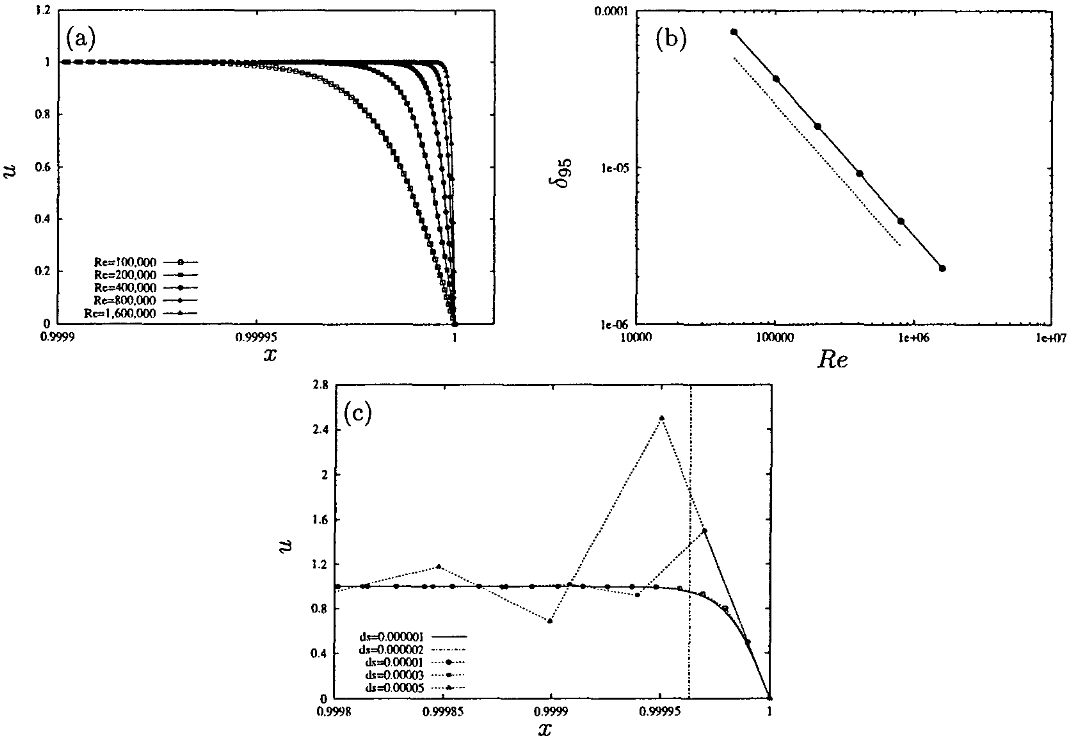

Figure B.2: (a) Velocity profile at various Reynolds numbers; (b) Dependence of boundary layer thickness δ_{95} on the Reynolds number; (c) Effects of grid size near the wall and onset of oscillations.

Bibliography

[1] F. Abergel and R. Temam. On some control problems in fluid mechanics. *Theoretical and Computational Fluid Dynamics*, 1:303–325, 1990.

[2] K. Akselvoll and P. Moin. Large-eddy simulation of turbulent confined coannular jets and turbulent flow over a backward facing step. Technical Report TF-63, Department of Mechanical Engineering, Stanford University, 1995.

[3] M. Alam and N. D. Sandham. Direct numerical simulation of short laminar separation bubbles with turbulent reattachment. *Journal of Fluid Mechanics*, 410:1–28, 2000.

[4] B. D. O. Anderson and J. B. Moore. *Optimal Filtering.* Prentice-Hall, Englewood Cliffs, New Jersey, 1979.

[5] B. Bamieh, F. Paganini, and M. Dahleh. Distributed control of spatially-invariant systems. *IEEE Transactions in Automatic Control*, 47:1091–1107, 2002.

[6] L. Baramov, O. R. Tutty, and E. Rogers. Robust control of linearized Poiseuille flow. *Journal of Guidance, Control and Dynamics*, 25(1):145–151, 2002.

[7] A. Y. Barraud. A numerical algorithm to solve $A^TXA - X = Q$. *IEEE Transactions in Automatic Control*, 22:883–885, 1977.

[8] T. Bewley, R. Temam, and M. Ziane. A general framework for robust control in fluid mechanics. *Physica D*, 138:360–392, 2000.

[9] T. R. Bewley. Flow control: new challenges for a new Renaissance. *Progress in Aerospace Sciences*, 37:21–58, 2001.

[10] T. R. Bewley and S. Liu. Optimal and robust control and estimation of linear paths to transition. *Journal of Fluid Mechanics*, 365:305–349, 1998.

[11] T. R. Bewley, P. Moin, and R. Teman. DNS-based predictive control of turbulence: an optimal benchmark for feedback algorithms. *Journal of Fluid Mechanics*, 447:179–225, 2001.

[12] J. P. Boyd. *Chebyshev and Fourier Spectral Methods.* Dover Publications Inc., New York, second edition, 1999.

[13] A. Brandt. Multilevel adaptive solutions to boundary-value problems. *Mathematics of Computation*, 31:333–390, 1977.

[14] M. Breuer. Large eddy simulation of the subcritical flow past a circular cylinder: numerical and modelling aspects. *International Journal for Numerical Methods in Fluids*, 28:1281–1302, 1998.

[15] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang. *Spectral Methods in Fluid Dynamics*. Springer-Verlag, New York, 1988.

[16] I. P. Castro and E. Epik. Boundary layer development after a separated region. *Journal of Fluid Mechanics*, 374:91–116, 1998.

[17] W. M. Chan and J. L. Steger. Enhancements of a three-dimensional hyperbolic grid generation scheme. *Applied Mathematics and Computation*, 51:181–205, 1992.

[18] R. C. Chang, F. B. Hsiao, and R. N. Shyu. Forcing level effects of internal acoustic excitation on the improvement of airfoil performance. *Journal of Aircraft*, 29(5):823–829, 1992.

[19] S. S. Collis. *A Computational Investigation of Receptivity in High-speed Flow Near a Swept Leading-edge*. PhD thesis, Stanford University, 1997.

[20] A. W. Cook and J. J. Riley. Direct numerical simulation of a turbulent reactive plume on a parallel computer. *Journal of Computational Physics*, 129:263–283, 1996.

[21] J. F. Donovan, L. D. Kral, and A. W. Cary. Active flow control applied to an airfoil. *AIAA paper* 98-0210.

[22] P. Van Dooren. A generalized eigenvalue approach for solving Riccati equations. *SIAM Journal on Scientific and Statistical Computing*, 2(2):121–135, 1981.

[23] J. K. Dukowicz and A. S. Dvinsky. Approximate factorization as a high order splitting for the implicit incompressible flow equations. *Journal of Computational Physics*, 102:336–347, 1992.

[24] G. E. Dullerud and F. Paganini. *A Course in Robust Control Theory: a Convex Approach*. Springer-Verlag, New York, 2000.

[25] J. H. Ferziger and M. Perić. *Computational Methods for Fluid Dynamics*. Springer-Verlag, Berlin Heidelberg, 1996.

[26] M. Gad-el-Hak. Control of low-speed airfoil aerodynamics. *AIAA Journal*, 28(9):1537–1552, 1990.

[27] M. Gad-el-Hak and D. M. Bushnell. Separation control: Review. *Journal of Fluids Engineering*, 113:5–30, 1991.

[28] M. Gaster. The structure and behavior of laminar separation bubbles. In *AGARD CP 4*, pages 813–854, 1966.

[29] M. Germano, U. Piomelli, P. Moin, and W. H. Cabot. A dynamics subgrid-scale viscosity model. *Physics of Fluids A*, 3(7):1760–1765, 1991.

[30] A. Glezer and M. Amitay. Synthetic jets. *Annual Review of Fluid Mechanics*, 34:503–529, 2002.

[31] G. H. Golub and C. F. Van Loan. *Matrix Computation*. John Hopkins University Press, Baltimore, Maryland, third edition, 1996.

[32] M. Green and D. J. N. Limebeer. *Linear Robust Control*. Prentice Hall, Englewood Cliffs, New Jersey, 1995.

[33] D. Greenblatt and I. J. Wygnanski. The control of flow separation by periodic excitation. *Progress in Aerospace Sciences*, 36:487–545, 2000.

[34] A. A. Hassan and R. D. JanakiRam. Effects of zero-mass synthetic jets on the aerodynamics of the NACA-0012 airfoil. *AIAA paper* 97-2326.

[35] C.-M. Ho and Y.-C. Tai. Micro-electro-mechanical-systems (MEMS) and fluid flows. *Annual Review of Fluid Mechanics*, 30:579–612, 1998.

[36] Y. Hoarau, M. Braza, Y. Ventikos, D. Faghani, and G. Tzabiras. Organized modes and the three-dimensional transition to turbulence in the incompressible flow around a NACA0012 wing. *Journal of Fluid Mechanics*, 496:63–72, 2003.

[37] M. Högberg, T. Bewley, and D. S. Henningson. Linear feedback control and estimation of transition in plane channel flow. *Journal of Fluid Mechanics*, 481:149–175, 2003.

[38] M. Högberg, T. Bewley, and D. S. Henningson. Relaminarization of Re_τ=100 turbulence using gain scheduling and linear state-feedback control. *Physics of Fluids*, 15(11):3572–3575, 2003.

[39] M. Högberg and D. S. Henningson. Linear optimal control applied to instabilities in spatially developing boundary layers. *Journal of Fluid Mechanics*, 470:151–179, 2002.

[40] H. P. Horton. *Laminar Separation in Two and Three-dimensional Incompressible Flow.* PhD thesis, University of London, 1968.

[41] K. Hsu and S. L. Lee. A numerical technique for two-dimensional mesh generation with grid control at all of the boundaries. *Journal of Computational Physics*, 96(2):451–469, 1991.

[42] R. Jacobs and P.A. Durbin. Simulations of bypass transition. *Journal of Fluid Mechanics*, 428:185–212, 2000.

[43] S. Jacobson and W. C. Reynolds. Active control of streamwise vortices and streaks in boundary layers. *Journal of Fluid Mechanics*, 360:179–211, 1998.

[44] S. Joshi, J. L. Speyer, and J. Kim. A systems theory approach to the feedback stabilization of infinitesimal and finite-amplitude disturbances in plane Poiseuille flow. *Journal of Fluid Mechanics*, 332:157–184, 1997.

[45] R. D. Joslin. Aircraft laminar flow control. *Annual Review of Fluid Mechanics*, 30:1–29, 1998.

[46] G. Kalitzin, X. Wu, and P. A. Durbin. DNS of fully turbulent flow in a LPT passage. *International Journal of Heat and Fluid Flow*, 24(4):636–644, 2003.

[47] S. Kang and H. Choi. Suboptimal feedback control of turbulent flow over a backward-facing step. *Journal of Fluid Mechanics*, 463:201–227, 2002.

[48] J. Kim. Control of turbulent boundary layers. *Physics of Fluids*, 15(5):1093–1105, 2003.

[49] J. Kim and J. Lim. A linear process in wall-bounded turbulent shear flows. *Physics of Fluids*, 12(8):1885–1888, 2000.

[50] J. Kim and P. Moin. Application of a fractional step method to incompressible Navier-Stokes equations. *Journal of Computational Physics*, 59:308–323, 1985.

[51] J. Kim, P. Moin, and R. Moser. Turbulence statistics in fully developed channel flow at low Reynolds number. *Journal of Fluid Mechanics*, 177:133–166, 1987.

[52] M. Kiya, M. Shimizu, and O. Mochizuki. Sinusoidal forcing of a turbulent separation bubble. *Journal of Fluid Mechanics*, 342:119–139, 1997.

[53] C. Lee, J. Kim, D. Babcock, and R. Goodman. Application of neural networks to turbulence control for drag reduction. *Physics of Fluids*, 9(6):1740–1747, 1997.

[54] C. Lee, J. Kim, and H. Choi. Suboptimal control of turbulent channel flow for drag reduction. *Journal of Fluid Mechanics*, 371:157–178, 1998.

[55] K. Lee, L. Cortelezzi, J. Kim, and J. Speyer. Application of reduced-order controller to turbulent flows for drag reduction. *Physics of Fluids*, 13(5):1321–1330, 2001.

[56] J. Lim and J. Kim. A singular value analysis of boundary layer control. *Physics of Fluids*, 16(6):1980–1987, 2004.

[57] L. Ljung. *System Identification.* Prentice Hall, Upper Saddle River, New Jersey, second edition, 1999.

[58] J. Macki and A. Strauss. *Introduction to Optimal Control Theory.* Springer-Verlag, New York, 1982.

[59] K. Mahesh, G. Constantinescu, and P. Moin. A numerical method for large-eddy simulation in complex geometries. *Journal of Computational Physics*, 197(1):215–240, 2004.

[60] W.J. McCroskey, K.W. McAlister, L.W. Carr, and S.L. Pucci. An experimental study of dynamic stall on advanced airfoil sections. NASA TM 84245, 1982.

[61] C. Min and H. Choi. Suboptimal feedback control of vortex shedding at low Reynolds numbers. *Journal of Fluid Mechanics*, 401:123–156, 1999.

[62] R. Mittal and S. Balanchandar. Effect of three-dimensionality on the lift and drag of nominally two-dimensional cylinders. *Physics of Fluids*, 7(8):1841–1865, 1995.

[63] R. Mittal and P. Moin. Suitability of upwind-biased finite difference schemes for large-eddy simulation of turbulent flows. *AIAA Journal*, 35(8):1415–1417, 1997.

[64] P. Moin and T. Bewley. Feedback control of turbulence. *Applied Mechanics Review*, 47(6):S3–S13, 1994.

[65] P. Moin and K. Mahesh. Direct numerical simulation: a tool in turbulence research. *Annual Review of Fluid Mechanics*, 30:539–578, 1998.

[66] Y. Morinishi, T. S. Lund, O. V. Vasilyev, and P. Moin. Fully conservative higher order finite difference schemes for incompressible flows. *Journal of Computational Physics*, 143:90–124, 1998.

[67] R. Moser, J. Kim, and M. M. Mansour. Direct numerical simulation of turbulent channel flow up to Re_τ=590. *Physics of Fluids*, 11(4):943–945, 1999.

[68] Y. Na and P. Moin. Direct numerical simulation of a separated turbulent boundary layer. *Journal of Fluid Mechanics*, 370:175–201, 1998.

[69] N. V. Nikitin, F. Nicoud, B. Wasistho, K. D. Squires, and P. R. Spalart. An approach to wall modeling in large-eddy simulation. *Physics of Fluids*, 12(7):1629–1632, 2000.

[70] P. Van Overschee and B. De Moor. *Subspace Identification for Linear Systems.* Kluwer Academic Publishers, 1996.

[71] P. R. Owen and L. Klanfer. On the laminar boundary layer separation from the leading edge of a thin airfoil. *ARC Conference Proceedings*, 220, 1953.

[72] L. L. Pauley, P. Moin, and W. C. Reynolds. The structure of two-dimensional separation. *Journal of Fluid Mechanics*, 220:397–411, 1990.

[73] C. D. Pierce. *Progress-variable Approach for Large-eddy Simulation of Turbulent Combustion.* PhD thesis, Stanford University, 2001.

[74] U. Piomelli, E. Balaras, H. Pasinato, K.D. Squires, and P.R. Spalart. The inner-outer layer interface in large-eddy simulations with wall-layer models. *International Journal of Heat and Fluid Flow*, 24:538–550, 2003.

[75] M. M. Rai and P. Moin. Direct simulations of turbulent flow using finite-difference schemes. *Journal of Computational Physics*, 96:15–53, 1991.

[76] R. Rathnasingham and K. S. Breuer. Active control of turbulent boundary layer. *Journal of Fluid Mechanics*, 495:209–233, 2003.

[77] M. Rosenfeld, D. Kwak, and M. Vinkour. A fractional-step solution method for the unsteady incompressible Navier-Stokes equations in generalized coordinate systems. *Journal of Computational Physics*, 94(1):102–137, 1991.

[78] P. J. Schmid and D. S. Henningson. *Stability and Transition in Shear Flows.* Springer-Verlag, New York, 2001.

[79] A. Seifert, T. Bachar, D. Koss, M. Shepshelovich, and I. Wygnanski. Oscillatory blowing: a tool to delay boundary-layer separation. *AIAA Journal*, 31(11):2052–2060, 1993.

[80] A. Seifert, A. Darabi, and I. Wygnanski. Delay of airfoil stall by periodic excitation. *Journal of Aircraft*, 33(4):691–698, 1996.

[81] A. Seifert and L. G. Pack. Oscillatory control of separation at high Reynolds numbers. *AIAA paper* 98-0214.

[82] M. Shur, P.R. Spalart, M. Strelets, and A. Travin. Detached-eddy simulation of an airfoil at high angle of attack. In W. Rodi and D. Laurence, editors, *Engineering Turbulence Modelling and Experiments 4*. Elsevier Science Ltd., 1999.

[83] P. R. Spalart and S. R. Allmaras. A one-equation turbulence model for aerodynamic flows. *AIAA paper* 92-0439.

[84] P. R. Spalart, W. H. Jou, M. Strelets, and S. R. Allmaras. Comments on the feasibility of LES for wings, and on a hybrid RANS/LES approach. In C. Liu and Z. Liu, editors, *Advances in DNS/LES*. Greyden Press, 1997.

[85] P. R. Spalart, R. D. Moser, and M. M. Rogers. Spectral methods for the Navier-Stokes equations with one infinite and two periodic directions. *Journal of Computational Physics*, 96:297–324, 1991.

[86] P. R. Spalart and M. K. Strelets. Mechanisms of transition and heat transfer in a separation bubble. *Journal of Fluid Mechanics*, 403:329–349, 2000.

[87] M. Stoke and D. S. Henningson. Direct numerical simulation of a separated turbulent boundary layer. *Journal of Fluid Mechanics*, 471:107–136, 2002.

[88] J. C. Tannehill, D. A. Anderson, and R. H. Pletcher. *Computational Fluid Mechanics and Heat Transfer*. Taylor and Francis, second edition, 1997.

[89] A. Travin, M. Shur, M. Strelets, and P. Spalart. Detached-eddy simulations past a circular cylinder. *Flow, Turbulence and Combustion*, 63:293–313, 1999.

[90] O. V. Vasilyev. High order finite difference schemes on non-uniform meshes with good conservation properties. *Journal of Computational Physics*, 157:746–761, 2000.

[91] M. Verhaegen. Identification of the deterministic part of MIMO state space models given in innovations form from input-output data. *Automatica*, 30(1):61–74, 1994.

[92] M. R. Visbal and D. P. Rizzetta. Large-eddy simulation on curvilinear grids using compact differencing and filtering schemes. *Journal of Fluids Engineering*, 124:746836–847, 2002.

[93] J. H. Watmuff. Evolution of a wave packet into vortex loops in a laminar separation bubble. *Journal of Fluid Mechanics*, 397:119–169, 1999.

[94] D. C. Wilcox. *Turbulence Modeling for CFD*. DCW Industries, Pasadena, California, 1998.

[95] J. Wu, X. Lu, A. Denny, M. Fan, and J. Wu. Post-stall flow control on an airfoil by local unsteady forcing. *Journal of Fluid Mechanics*, 371:21–58, 1998.

[96] K. Zhou, J. Doyle, and K. Glover. *Robust and Optimal Control*. Prentice Hall, Upper Saddle River, New Jersey, 1996.